



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**REAL-TIME DATA ACQUISITION AND PROCESSING OF  
THE MAGNETIC, ANGULAR RATE AND GRAVITY  
(MARG) SENSOR**

by

Ioannis R. Saliaris

June 2004

Thesis Advisor:  
Second Reader:

Xiaoping Yun  
David C. Jenn

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> June 2004	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE:</b> Real-Time Data Acquisition and Processing of the Magnetic, Angular Rate and Gravity (MARG) Sensor			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Ioannis R. Saliaris				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> U.S. Army Research Office (ARO) U.S. Navy Modeling and Simulation Office (N6M)			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b> <p>This research involves the development of a human-body motion tracking system constructed with the use of commercial off-the-shelf (COTS) components. The problem to be solved is that the data from the motion tracking sensors must be transmitted wirelessly in real time from a microcontroller to a server computer. Due to the fact that the microcontroller does not support a standard OS, widely used PCMCIA cards or USB wireless modules cannot be used. The wireless communication module chosen for this purpose is the DPAC <i>airborne</i>, a highly integrated 802.11b module that can be easily integrated with the microcontroller.</p> <p>The evaluation of the module was completed in four stages. The first part was to initiate communication with the DPAC module. The second part was to establish communication between the DPAC module and a TCP server. The third part was to establish communication between the microcontroller and the DPAC module. The fourth part was to increase the baud-rate to the desired high value of 230,400 bps.</p> <p>The evaluation result indicates that the DPAC <i>airborne</i> module meets the wireless communication requirements of the motion tracking system.</p>				
<b>14. SUBJECT TERMS</b> MARG III Sensor, Control Interface Unit (CIU), Microcontroller, RS232, UART, PCMCIA, USB, Wireless Communication, 802.11, 802.11b, WiSER 2400, DPAC <i>airborne</i> .			<b>15. NUMBER OF PAGES</b> 84	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b>  Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b>  Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b>  Unclassified	<b>20. LIMITATION OF ABSTRACT</b>  UL	

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for Public Release; Distribution is Unlimited**

**REAL-TIME DATA ACQUISITION AND PROCESSING OF THE MAGNETIC,  
ANGULAR RATE AND GRAVITY (MARG) SENSOR**

Ioannis R. Saliaris  
Lieutenant, Hellenic Navy  
B.S., in Naval Science, Hellenic Naval Academy, 1996  
B.S., Equivalency in Electrical Engineering, Naval Postgraduate School, 2002

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING  
and  
MASTER OF SCIENCE IN SYSTEMS ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL  
June 2004**

Author: Ioannis R. Saliaris

Approved by: Xiaoping Yun  
Thesis Advisor

David C. Jenn  
Second Reader

John P. Powers  
Chairman, Department of Electrical and Computer Engineering

Dan C. Boger  
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

## ABSTRACT

This research involves the development of a human-body motion tracking system constructed with the use of commercial off-the-shelf (COTS) components. The problem to be solved is that the data from the motion tracking sensors must be transmitted wirelessly in real time from a microcontroller to a server computer. Due to the fact that the microcontroller does not support a standard OS, widely used PCMCIA cards or USB wireless modules cannot be used. The wireless communication module chosen for this purpose is the DPAC *airborne*, a highly integrated 802.11b module that can be easily integrated with the microcontroller.

The evaluation of the module was completed in four stages. The first part was to initiate communication with the DPAC module. The second part was to establish communication between the DPAC module and a TCP server. The third part was to establish communication between the microcontroller and the DPAC module. The fourth part was to increase the baud-rate to the desired high value of 230,400 bps.

The evaluation result indicates that the DPAC *airborne* module meets the wireless communication requirements of the motion tracking system.

THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	PREVIOUS WORK.....	1
B.	RESEARCH ISSUES .....	2
C.	THESIS GOALS.....	4
D.	THESIS ORGANIZATION OUTLINE .....	4
II.	DESCRIPTION OF THE TI MSP430F149 ULTRA-LOW POWER MICROCONTROLLER .....	7
A.	THE TI MSP430F149 MICROCONTROLLER .....	8
B.	UNIVERSAL ASYNCHRONOUS COMMUNICATION INTERFACE (UART) .....	9
C.	THE UART ON THE TI MICROCONTROLLER.....	10
D.	SUMMARY .....	11
III.	SELECTION OF WIRELESS COMMUNICATION COMPONENTS FOR THE MARG PROJECT .....	13
A.	REQUIREMENTS.....	13
B.	WIRELESS LAN STANDARDS .....	14
1.	The IEEE 802.11 Standard.....	15
2.	The 802.11b Standard.....	16
3.	The 802.11a Standard.....	17
4.	The 802.11g Standard.....	18
5.	Other Members of the 802.11 Family .....	19
6.	Deciding in Favor of the 802.11b Protocol .....	20
C.	COMMERCIAL-OFF-THE-SHELF (COTS) WLAN COMPONENTS .....	20
1.	The Use of a Wireless PCMCIA Card .....	21
2.	The Use of a Wireless USB Device.....	21
3.	The Use of a RS232 Wireless Communication Device ..	22
4.	The Use of a Multi-purpose Wireless Communication Device .....	23
D.	THE DPAC <i>AIRBORNE</i> WIRELESS COMMUNICATION UNIT ..	24
E.	THE DPAC <i>AIRBORNE</i> WLAN NODE MODULE EVALUATION AND DEVELOPMENT KIT.....	25
F.	SUMMARY .....	27
IV.	TESTING AND EVALUATION.....	29
A.	COMMUNICATING WITH THE DPAC <i>AIRBORNE</i> .....	29
1.	Setting up the DPAC <i>Airborne</i> .....	31
2.	Using the Web Browser .....	34
B.	ESTABLISHING COMMUNICATION BETWEEN THE AIRBORNE WIRELESS COMMUNICATION MODULE AND A TCP SERVER .....	36

C.	ESTABLISHING COMMUNICATION BETWEEN THE <i>AIRBORNE</i> WIRELESS COMMUNICATION UNIT AND THE TI MICROCONTROLLER.....	37
D.	INCREASING THE BAUD-RATE.....	39
1.	Using the Stock Crystal (32,768 Hz).....	39
2.	Using an Aftermarket Crystal (7,372,800 Hz).....	41
E.	SUMMARY .....	42
V.	LINKING THE TI MSP430F149 AND THE DPAC <i>AIRBORNE</i> COMMUNICATION UNIT.....	43
A.	THE DESIGN OF THE DPAC <i>AIRBORNE</i> MODULE BOARD.....	43
B.	THE FABRICATION OF THE DPAC <i>AIRBORNE</i> MODULE BOARD .....	45
C.	THE EVALUATION AND TESTING OF THE DPAC <i>AIRBORNE</i> MODULE BOARD AND ITS CONNECTION WITH THE TI MICROCONTROLLER.....	46
D.	SUMMARY .....	49
VI.	CONCLUSIONS – FURTHER DEVELOPMENT .....	51
A.	THESIS CONTRIBUTIONS.....	51
B.	SUGGESTIONS FOR FURTHER DEVELOPMENT .....	52
	APPENDIX: THE MATLAB PROGRAM FOR THE CALCULATION OF THE MODULATION DATA SHIFT REGISTER .....	55
	LIST OF REFERENCES.....	59
	INITIAL DISTRIBUTION LIST .....	61

## LIST OF FIGURES

Figure 1.	The TI Microcontroller.....	7
Figure 2.	The TI Microcontroller Architecture [From Ref. 9].....	9
Figure 3.	The UART Mode Block Diagram [From Ref. 9].....	10
Figure 4.	The NETGEAR Wireless USB Network Adapter [From Ref. 17].	21
Figure 5.	The OTC WiSER Wireless Serial RS232 Network Adapter [From Ref. 18].....	22
Figure 6.	The DPAC WLAN Module [From Ref. 19].....	24
Figure 7.	The DPAC <i>Airborne</i> Hardware Block Diagram [From Ref. 7].....	25
Figure 8.	The DPAC <i>Airborne</i> Evaluation Board Layout [After Ref. 21]....	26
Figure 9.	The DPAC <i>Airborne</i> Evaluation Board.....	26
Figure 10.	Hardware Configuration [From Ref. 20].....	30
Figure 11.	The Hardware Configuration Used [After Ref. 20]. ....	31
Figure 12.	Verifying the “Remote Computer’s” IP Address.....	32
Figure 13.	The Serial Port Connector on the DPAC <i>Airborne</i> EVB. ....	33
Figure 14.	Connecting to the DPAC <i>Airborne</i> Using Internet Explorer.....	35
Figure 15.	Web Browser Instances. ....	36
Figure 16.	The TI Microcontroller – DPAC <i>Airborne</i> Connection. ....	38
Figure 17.	The External Crystal of the TI Microcontroller. ....	40
Figure 18.	The Schematic of the DPAC <i>Airborne</i> Module Board Layout (not to scale).....	44
Figure 19.	The DPAC <i>Airborne</i> Module Board Layout.....	45
Figure 20.	The DPAC <i>Airborne</i> Module Board. ....	46
Figure 21.	The DPAC <i>Airborne</i> Module Board. ....	47
Figure 22.	The DPAC <i>Airborne</i> Module Board Linked to the TI Microcontroller of the CIU.....	48
Figure 23.	The Optimized DPAC <i>Airborne</i> Module Board Layout. ....	49
Figure 24.	The MATLAB Window Screenshot Showing the Output of the Program.....	57

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF TABLES

Table 1.	HyperTerminal Configuration for the Initial DPAC <i>Airborne</i> EVB Set-up. ....	33
----------	---	----

THIS PAGE INTENTIONALLY LEFT BLANK

## **ACKNOWLEDGMENTS**

This project is in part supported by the U.S. Army Research Office (ARO) and the U.S. Navy Modeling and Simulation Office (N6M).

I certainly owe thanks to my family, friends and especially to my wife for their support.

The people who can never be thanked enough for their contribution are the teachers, to whom I am indebted. Yet, here I would like to acknowledge and express my absolute gratitude to the teacher who happened to be my thesis advisor, namely Professor Xiaoping Yun. His brilliance illuminated the paths where I was lead by the research procedure, and his tireless support helped me conquer the goals preset. His managerial abilities were apparent and an experienced engineer guided me systematically toward the goal. The result was the successful completion of my thesis.

At this point, I owe to acknowledge Professor Robert G. Hutchins, whom I am lucky to have had as the teacher who taught me most of the courses in the “Guidance, Control, and Navigation Systems Option.”

I also owe thanks to James Calusdian for being there every time I needed his precious knowledge and experience in laboratory practicalities.

I would also like to thank Ron Russell for his meticulous, speedy and on the subject editing of this thesis. His efficient work was extremely helpful.

THIS PAGE INTENTIONALLY LEFT BLANK

*"I find that the harder I work the more luck I seem to have."*

- Thomas Jefferson (1743 -1826)

THIS PAGE INTENTIONALLY LEFT BLANK

*To Anastasia*

THIS PAGE INTENTIONALLY LEFT BLANK

## EXECUTIVE SUMMARY

This research involves the development of a human-body motion tracking system constructed with the use of commercial off-the-shelf (COTS) components. The MARG motion tracking system has direct military applications, including the implementation of a Synthetic Environment for Virtual Combat Training purposes. The MARG system consists of fifteen sensors responsible for sensing rotational motions, the Control Interface Unit (CIU) responsible for gathering and forming packets from the data coming from the sensors, the wireless communication module (DPAC *airborne*) responsible for wirelessly transmitting the data, and finally the server computer responsible for depicting the motion.

The main component of the system investigated in this thesis is the wireless communication module that sends the motion data to a server computer. The wireless communication module chosen for this purpose is the DPAC *airborne*, a highly integrated 802.11b module fulfilling all the performance requirements for the MARG project. It is configured to receive the data from the TI microcontroller of the CIU and retransmit them to any client participating in the Wireless LAN.

The evaluation of the module was completed in four stages. The first part was to initiate communication with the DPAC module. The second part was to establish communication between the DPAC module and a TCP server. The third part was to establish communication between the microcontroller and the DPAC module. The fourth part was to increase the baud-rate to the desired high value of 230,400 bps.

Based on the valuation conducted, it is determined that the DPAC *airborne* meets all the requirements of the MARG project. In particular it achieves the 230,400 bps data rate and is easily integrated into the CIU.

THIS PAGE INTENTIONALLY LEFT BLANK

## **I. INTRODUCTION**

In this chapter, the concept of the MARG project is presented. After a brief discussion of the previous work on the project, the research issues addressed in this thesis are introduced. In addition, the contents of the remaining thesis chapters are outlined.

### **A. PREVIOUS WORK**

The MARG project originates from the idea of representing the motion of the human body fully and in real time [Ref. 1]. This motion is represented in a virtual environment, which is also known as a Synthetic Environment [Ref. 2]. The design goal of such an environment is to make the user feel as if he exists in that environment [Ref. 1]. The military relevance of the Synthetic Environment is its use for Virtual Combat Training purposes.

Bachman [Ref. 3] made the first coherent attempt to create such a virtual environment. For his dissertation, he used the second generation of the prototype sensors MARG (Magnetic, Angular Rate, and Gravity), MARG II. One MARG sensor was placed on each (human) limb. The intention was for every sensor to monitor the three-degrees-of-freedom motion of each limb.

The MARG II sensors are comprised of three magnetometers, three accelerometers, and three angular rate micro-machined sensors [Ref. 1]. They were physically connected to a central computer. There the analog data was gathered and filtered so that the limb motion could be depicted on an avatar.

Since that time, the virtual environment has evolved. In particular the following drawbacks were addressed. First, the power supply for each of those sensors was a 12-Volt DC battery. As a result, the sensors were bulky and heavy [Ref. 1].

Second, the power consumption of the sensors was quite high, requiring one to carry along sufficient battery cells [Ref. 1].

Third, the previously mentioned central computer was specially configured for the project. This set up was arduous and involved various bulky hardware. As a result, the central computer constituted a single point of failure. In addition, replacing it was a strenuous procedure [Ref. 1].

Fourth, the sensors were physically attached (cabled) to the central computer [Ref. 1].

Kavousanos-Kavousanakis [Ref. 1] made the second coherent attempt to create a virtual environment designed with the above drawbacks in mind. Since then, the MARG III sensor had been developed. This sensor is smaller, programmable, and consumes less power.

A Control Interface Unit (CIU) was introduced. One goal of the CIU was to reduce some of the processing load from the central computer. However its main goal was to gather and to prepare the digital data to be transmitted serially through a wireless interface. An additional task of the CIU was to supply power to the sensors. In that way the weight of the batteries that comprise the power supply could be transferred from the sensors to the CIU [Ref. 1].

The central computer was replaced by a standard PC. The only requirement for that PC would be to run a server program with the CIU as its client.

## **B. RESEARCH ISSUES**

The MARG III version of the project revealed some drawbacks in its implementation. The commercial-off-the-shelf (COTS) wireless communication device that was used demanded an RS232-compatible input from the CIU. That resulted in several problems. First, the RS232 protocol is a high-voltage protocol, which means it has a relatively high power consumption. It uses  $\pm 15$  Volt DC [Ref. 4]. The rationale is that RS232 was designed to perform adequately over longer distances. However this feature is unneeded for this application since the maximum distance to be covered is on the order of the size of the human body, merely a few feet.

Second, the maximum bit-rate supported by the RS232 protocol for commercially designed products is 115,200 bits per second (bps) [Ref. 5]. This bit-rate is exactly the half of what is needed with all fifteen sensors in operation. This last statement considers the sensor data rate, and the sensor-to-CIU and CIU-to-serial interface. The incoming data rate consists of the sum of the data rate for each sensor, plus the overhead bits used in the sensor-to-CIU – CIU-to-serial interface. As a result, the final figure describing the incoming data rate is 230,400 bps. When a lower data rate is used, the additional bits arriving must be buffered until they can be transmitted. Therefore, it is obvious that, if the bit-rate is kept at 115,200 bps, the only option is to sacrifice the real-time transmission and to introduce buffering.

The following points summarize the challenges that must be overcome in order to make the MARG project work in full (fifteen sensors), without having to address buffering issues:

- Increase the maximum bit-rate potential in the communication between the CIU and the wireless communication device,
- Employ an easily programmable wireless communication device that supports data rates of 230,400 bps or higher (in order to be able to meet future demands),
- Reduce the output voltage from the CIU in order to decrease further power consumption.

In order to resolve the above challenges, an in-depth search was initiated to find the proper components to integrate into the MARG III project. These components should support multiple protocols, which can be interfaced with the existing CIU, specifically, with the TI MSP430F149 microcontroller [Ref. 6] used in the CIU.

### C. THESIS GOALS

The main goal of this thesis was to integrate into the project a viable high-speed connection between the CIU and the wireless communication unit. The following had to be achieved in order to overcome real-time and buffering issues:

- Find a communication protocol that supports data rates of 230,400 bps and higher, and whose interface is simple enough to be used with a microcontroller like the TI MSP430F149,
- Find a wireless communication unit that supports that protocol,
- Program the microcontroller to use that protocol,
- Connect and interface the microcontroller with the communication unit,
- Test various speeds and set-ups using the evaluation boards of the microcontroller and the wireless communication device, and finally
- Integrate the wireless communication unit with the MARG motion tracking system.

### D. THESIS ORGANIZATION OUTLINE

Chapter II presents the TI MSP430F149. The main features of the TI MSP430F149 are briefly described. The particularities concerning the microcontroller are addressed.

Chapter III presents the DPAC *airborne* wireless communication unit [Ref. 7]. A brief discussion explains the reasons this communication unit was chosen over the existing and operating *WiSER* communication unit. The particularities concerning the unit are addressed.

Chapter IV presents the steps taken to evaluate the *airborne* unit. The most important part of that process – the evaluation and testing of the connection established between the TI MSP430F149 and the *airborne* wireless communication unit – is briefly discussed.

Chapter V describes the procedure followed to set up the microcontroller as well as the communication unit. In addition, it illustrates the steps followed in order to connect them and to initiate the communication.

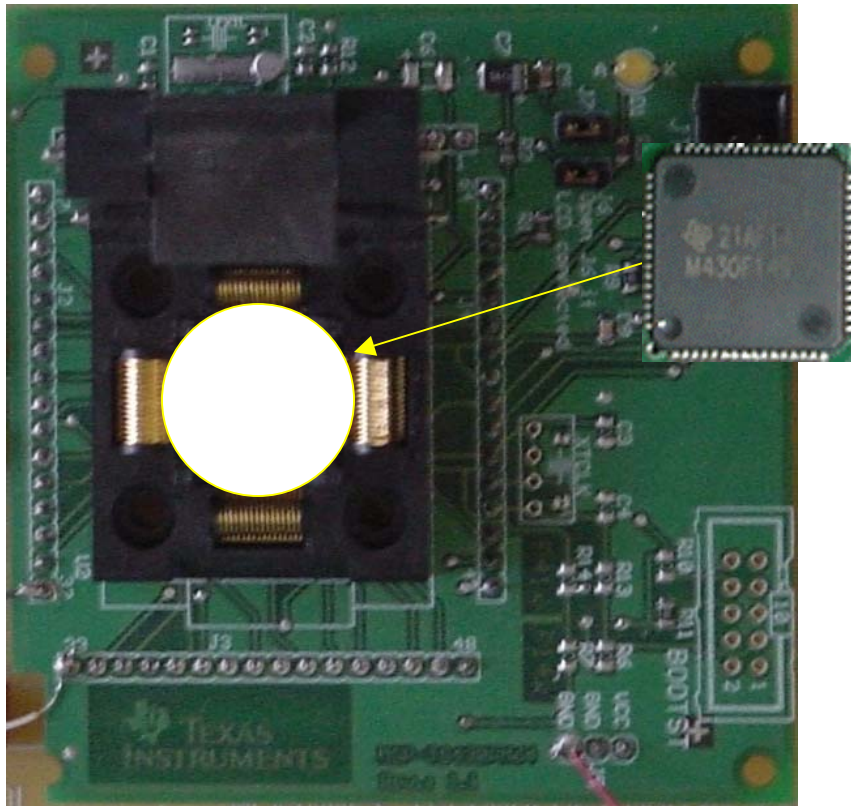
The final chapter (VI) of this thesis presents conclusions and suggestions to further develop and optimize the results.

THIS PAGE INTENTIONALLY LEFT BLANK

## II. DESCRIPTION OF THE TI MSP430F149 ULTRA-LOW POWER MICROCONTROLLER

This chapter provides a brief discussion on the TI MSP430F149 ultra-low power microcontroller (TI microcontroller). The TI microcontroller (an important part of the CIU) is the last component before the wireless transmission of the data gathered by the sensors. In the CIU, it collects the data from the sensors (after they are multiplexed) and assembles them in packets, ready for transmission [Ref. 1]. The data is then sent to the wireless communication device. For the purpose of this thesis, the TI microcontroller must be programmed to send the data via the appropriate high-speed protocol to the wireless communication device.

Earlier in the project, and after an extensive trade-off analysis, the TI microcontroller depicted in Figure 1 was selected.



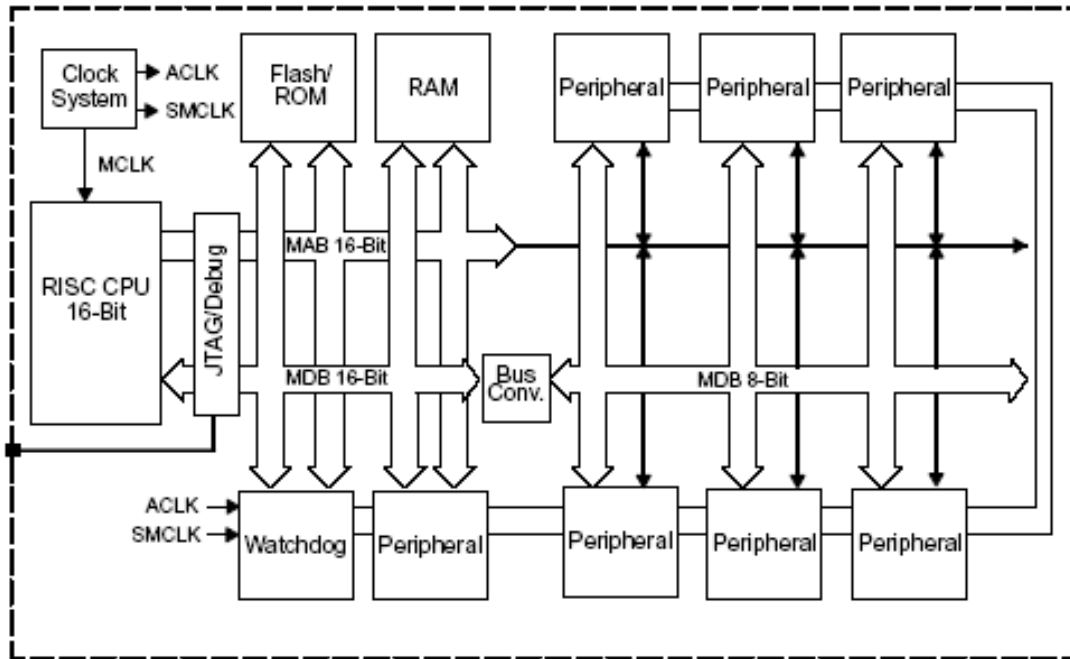
**Figure 1. The TI Microcontroller.**

The TI microcontroller features a powerful 16-bit RISC CPU with 16-bit registers and constant generators [Ref. 6]. Its most important feature though (for this thesis) is the universal serial synchronous/asynchronous communication interface (USART).

#### **A. THE TI MSP430F149 MICROCONTROLLER**

The TI MSP430F149 microcontroller features a RISC CPU (16-bit), some peripherals, and a clock system [Ref. 9]. The interconnections are made with a von-Neumann common memory address bus (MAB) and memory data bus (MDB) [Ref. 9]. The TI microcontroller therefore combines a modern CPU with modular analog and digital peripherals and offers solutions for demanding mixed-signal applications [Ref. 9].

The TI microcontroller's clock system includes an integrated high-speed digitally controlled oscillator (DCO), which can be a source for the master clock (MCLK) of the TI microcontroller. In addition, an auxiliary clock is driven directly from a common crystal [Ref. 9]. Any of the above-mentioned clocks can be used for synchronization purposes. Figure 2 shows a schematic of the architecture of the TI microcontroller.



**Figure 2. The TI Microcontroller Architecture [From Ref. 9].**

## **B. UNIVERSAL ASYNCHRONOUS COMMUNICATION INTERFACE (UART)**

UART is the asynchronous mode of USART. It is the component that handles asynchronous serial communication. The UART takes the bytes of data to be transmitted and transmits them as individual bits in a sequential fashion. At the destination, a second (receiver) UART reassembles the bits back into the complete bytes [Ref. 10].

A key feature of UART is that it allows data to be sent without the need for a clock signal. Instead of a clock, various timing parameters are agreed upon in advance (selected baud-rate). In addition there are special bits added to each word, which are used for synchronization purposes [Ref. 10].

The data rate is not limited as with the RS232 protocol to 115,200 bps. In addition, the voltages used are not as high.

### C. THE UART ON THE TI MICROCONTROLLER

The TI microcontroller supports three USART interfaces: UART, SPI, and I2C. In the UART mode, the TI microcontroller connects to an external properly interfaced subsequent unit via two external pins, URXD and UTXD [Ref. 9]. The UART mode block diagram is shown in Figure 3.

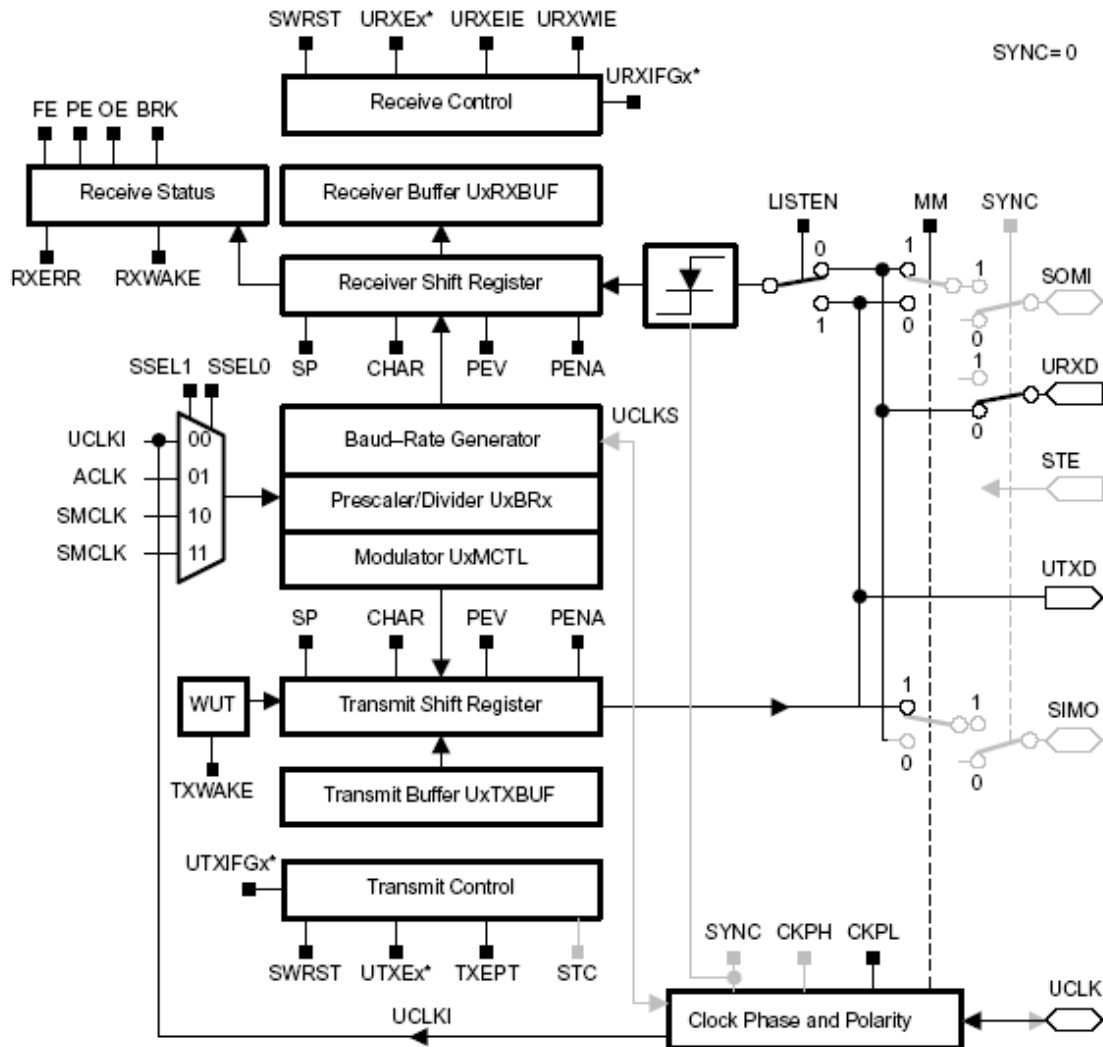


Figure 3. The UART Mode Block Diagram [From Ref. 9].

The UART mode in the TI microcontroller is used for the communication of the CIU and the wireless communication unit. As mentioned above, the timing for the UART mode is provided by the selected baud-rate. Furthermore, the clock/

crystal can be selected among the available clocks of the TI microcontroller. In this thesis, a high frequency crystal is used to achieve a high baud-rate in order to keep constant the desired baud-rate.

#### **D. SUMMARY**

In this chapter, some of the basic features of the TI microcontroller were described. Furthermore, those features that will be exploited in this thesis are pointed out. Then the UART interface was briefly discussed as well as the UART capabilities of the TI microcontroller.

The next chapter discusses how the selection of particular wireless communication components for the MARG project was made.

THIS PAGE INTENTIONALLY LEFT BLANK

### **III. SELECTION OF WIRELESS COMMUNICATION COMPONENTS FOR THE MARG PROJECT**

Among the goals of the MARG project is to make the MARG motion-tracking system wireless. There are various wireless communication standards and products available from which to choose.

This chapter first presents the wireless communication requirements of the MARG system, the 802.11 wireless standards family, and the commercial-off-the-shelf wireless components that could be integrated into the MARG system. It then discusses reasons that a particular 802.11b wireless communication unit, namely the DPAC *airborne*, is selected for the MARG system. It finally provides a brief description of the DPAC *airborne*.

The DPAC *airborne* is the component that is responsible for the wireless transmission (implementing the 802.11b standard) of the collected data from the CIU to a server (on a remote PC). For the purpose of this thesis, the DPAC *airborne* will have to be set up to receive the data via the appropriate protocol. As discussed in the previous chapter, the appropriate protocol is UART.

#### **A. REQUIREMENTS**

There are a few unyielding wireless communication requirements for the MARG system. First, the MARG system imposes the hardware interface requirement, which dictates that the form factor selected for the wireless transmission must support either RS232, SPI, UART or any other basic protocol that is easily implemented on a low-cost microcontroller, such as the TI microcontroller used in the MARG system.

Second, there is a software interface requirement since the MARG system cannot afford an Operating System (OS) such as Windows nor Windows-based drivers. The processing unit (TI microcontroller) is not designed for use with a standard OS.

Third, a quite high data rate is required for the connection between the microcontroller and the wireless communication unit in order for the data from all fifteen sensors of the MARG system to be sent in real time. The minimum acceptable data rate is 230,400 bps (as mentioned in Chapter II), which exactly covers the needs of fifteen sensors.

Other requirements are the need for low-power consumption, small size and finally low cost.

## **B. WIRELESS LAN STANDARDS**

For the MARG project, the use of the 802.11b protocol was the obvious solution from the start. It constitutes an easy and popular way to communicate data wirelessly from the human carrier of the sensors to a PC. Nevertheless, it is not the only available solution. Other solutions to the wireless communication problem are the *Bluetooth* technology and of course the other protocols that form the IEEE 802.11 family.

A brief description of the alternatives is provided to better understand the reason that an 802.11b standard was chosen.

The *Bluetooth* technology was discarded as a solution almost immediately, mainly because of its limited range (about 10 meters). It is true that although a ten-meter range is adequate for wireless personal area networking applications [Ref. 11], in this case it is far from sufficient. Ten meters from a stationary access point would not suffice even for demonstration purposes.

The next step was to look into the IEEE 802.11 family of wireless communication protocols. The following subsections describe the main members of the family, followed by the reasons the 802.11b protocol comprises the best solution.

## **1. The IEEE 802.11 Standard**

In 1980, the Institute of Electrical and Electronics Engineering (IEEE) started a project to establish the standards for Local Area Networks (LANs) and Metropolitan Area Networks (MANs) [Ref. 12].

In 1997, the original 2.4 GHz wireless Ethernet standard 802.11 was published [Ref. 13]. The data transfer rate for this standard was 1 to 2 Mbps. The transmission technique schemes selected were Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) [Ref. 14].

Spread-spectrum technology (which started being used commercially in the late 1980's) is a favorite technology of the armed forces because of its resistance to jamming as well as to interception [Ref. 15]. The spread-spectrum technology works as follows. The original signal is distributed into multiple frequencies (carriers), and each frequency bears part of the information and part of the original's signal power. Because the components are numerous and spread in the frequency spectrum, each component has power that practically lies below the noise level. Theoretically, only the receiver can isolate the various component frequencies comprising the signal from the noise and reassemble them into the original signal. From the above, it is obvious that the spread-spectrum technology is of great commercial value since it provides the ability to increase the efficiency of a very popular frequency band.

The result from IEEE was the generation of a very efficient, open standard that enabled the connection of PCs, PDAs, and other communication devices to Wireless LANs (WLANs) [Ref. 14]. The 802.11 achieved all that without creating much interference in this otherwise overpopulated frequency band, but most importantly, being relatively immune to interference itself. Yet, this standard was only the beginning. Its characteristics and low-data rate were due mainly to the particular combination of frequency, bandwidth, and modulation that were selected and implemented [Ref. 12]. Through a modification and perturbation of these factors, the other protocols of the family came into being. It is noted that

the original 2-Mbps 802.11 wireless communication standard still constitutes the fall back for the newer standards under difficulties or conflicts [Ref. 13].

## **2. The 802.11b Standard**

The next wireless communication standard to be implemented was 802.11b. The 802.11b standard presented a data rate of 11 Mbps operating in the 2.4 GHz band like 802.11. It quickly became the most popular and most used wireless communication standard [Ref. 13]. For a while, it was also known as the Wireless Fidelity Standard (Wi-Fi) [Ref. 13]. However, today the term refers to the whole 802.11 family of standards [Ref. 16].

The 802.11b standard can be thought of as the evolution of 802.11. The reason for this is that 802.11b uses not only the same frequency band as 802.11, but also uses a diversification of the DSSS scheme used in 802.11. The higher data rates accomplished are a result of the spread-spectrum modulation techniques used. [Ref. 12]

The 802.11b standard owes its significant commercial success to several causes. First, it uses the frequency band of 2.4 GHz, so all previous infrastructures can be used and in addition no license is required.

Second, implementing the modulation techniques used in the 802.11b standard is relatively simple and not as power consuming [Ref. 12].

Third, the 802.11b standard has the potential of transferring data to distances up to several hundred feet indoors and up to ten miles outdoors (line of sight) [Ref. 12].

Of course, there are also disadvantages to this standard. The disadvantages also are closely related to the frequency band selected. The frequency band used is relatively low in the wireless communication spectrum. Given the fact that lower frequencies have narrower usable bandwidth than higher frequencies, there are physical bandwidth limitations in comparison with other higher frequency standards. In addition, the 2.4 GHz band is very popular and this fact

makes it more probable for any device operating there to experience interference, even when using spread-spectrum techniques. [Ref. 12]

### **3. The 802.11a Standard**

The first protocol that appeared after the initial 802.11 was the 802.11a. Yet it took a long time (and in particular after the 802.11b was already being used) before it was actually implemented. With the 802.11a, a somewhat different approach was attempted than for the 802.11. Since the main drawback of the 802.11 was its slow data rate, the new attempt focused on increasing the data transfer speed. The 802.11a was able to provide up to 54 Mbps. The frequency used was the 5-GHz band (Unlicensed National Information Infrastructure (UNII) band). In addition, rather than using FHSS or DSSS, an orthogonal frequency division multiplexing (OFDM) encoding scheme was used [Ref. 14].

The choice of the UNII band to increase the data rate differentiates 802.11a from 802.11 as well as from its descendants (802.11b). However, this choice also introduced two new problems. The first of the two major problems was that a higher frequency is bound to decrease the maximum achievable distance (given the same transmitter power) due to higher free space path loss and attenuation through material. Second and most importantly, this high frequency increases the multi-path fading. In an effort to overcome this kind of problem, the coded OFDM (COFDM) encoding technique (developed from OFDM specifically for wireless indoor use) was chosen over the FHSS and DSSS [Ref. 12].

COFDM divides the bandwidth (20 MHz) of the high-speed data carrier into a number of low-speed (~ 300 KHz) sub-carriers [Ref. 12]. Most of these sub-channels are used for the transmission of the actual data while the rest are used for error correction [Ref. 12].

The conclusion is that 802.11a was not the proper second step because it changed the frequency band used, wasting the whole infrastructure based on the

802.11 standard. In addition, it was a relatively difficult standard to implement, so it did not become very popular although it provided a relatively high data rate [Ref. 12].

#### **4. The 802.11g Standard**

The 802.11b standard, as mentioned before, was very successful. Its disadvantages related to the 2.4-GHz frequency band cannot be addressed since the frequency band that causes some disadvantages is also one of the main reasons for its notable success. Consequently, the attribute of the 802.11b standard that could be enhanced was the transmission speed, the 11-Mbps data rate. This was the goal of the 802.11g standard. The specification for the 802.11g standard was the use of the 2.4 GHz frequency band and a data rate of 22 Mbps extendable to 54 Mbps [Ref. 12]. For the specifications to be met, more sophisticated transmission techniques had to be used. The IEEE task group decided to use the transmission technique scheme used in the 802.11a standard (OFDM) [Ref. 12].

The central idea behind the 802.11g standard was to gather the advantages of the 802.11a standard and include them with the features that made the 802.11b successful. The 802.11g standard was published in June 2003 [Ref. 12]. One of the main advantages of the new standard, besides the advantages it shares with 802.11b and its high data rate, is its backward compatibility with the 802.11b standard. This particular advantage will make the transition from one standard to the other easier.

The 802.11g shares exactly the same disadvantages as its 802.11b predecessor. However, it supports higher data rates and claims to achieve somewhat improved data transmission range [Ref. 14].

Moreover, a point about the 802.11g standard must be made. Although the 802.11b products will dominate the WLAN market over the next several years as expected, the 802.11g is to be considered their long-term successor [Ref. 12]. Most probably the 802.11g standard products will simultaneously support the

802.11a standard, thereby resulting in dual-band modules on both 2.4 and 5 GHz. [Ref. 12]. Therefore, the resources invested in an infrastructure supporting either the 802.11a or the 802.11b will not be wasted, at least not for the near future.

## **5. Other Members of the 802.11 Family**

Besides the popular 802.11a, b and g, there are a few others, the most important of which are presented below [Ref.13]:

- 802.11e: The 802.11e standard intends to improve the quality of service (QoS) by allowing packets with specific requirements (in transmission delay and bandwidth) to be transmitted in preference to packets with less restrictive requirements. When this standard is completed and available, it will improve streaming audio and video and work with already existing wireless cards.
- 802.11f: Yet another unfinished standard, which intends to allow users to move through a WLAN and maintain their connection, even when there are multiple access points from various manufacturers.
- 802.11h: The “h” letter in 802.11h stands for Hiperlan, which is the European WLAN standard. The 802.11h standard actually is a diversification of the 802.11a standard modified to be suitable for use in Europe and in order to be Hiperlan (modifications include frequency and power management issues in order to avoid interference with satellite and radar frequencies).
- 802.11i: This standard intends to take the 802.11 family to the next level of security, through key management and distribution, encryption and authentication. When the standard is finished, it will probably be integrated into the existing systems through a firmware upgrade.

- 802.11 IR: The 802.11 IR standard is the infrared (IR) version of 802.11. It was developed at the same time as the 802.11 and has the same characteristics and data rate. This standard was never implemented.

## **6. Deciding in Favor of the 802.11b Protocol**

From the above discussion on the various standards, it is not hard to recognize that the 802.11g standard is the best yet. Nevertheless, it was decided not to use the 802.11g standard in the MARG project for the following reasons. First, no 802.11g compatible wireless communication module serving the project purpose could be found. Although there are a variety of 802.11g modules, none supports the stand-alone features, integration potential, and flexibility (e.g. High speed UART) required.

Second, there is an existing infrastructure throughout the NPS campus, supporting the 802.11b WLAN standard.

Third, In the case of the MARG project, the data rate provided by the 802.11b standard is presently sufficient. In addition, because of the backward compatibility provided by the 802.11g standard, even if the future of the wireless communications is the 802.11g standard, the MARG project components will continue to operate.

## **C. COMMERCIAL-OFF-THE-SHELF (COTS) WLAN COMPONENTS**

From the beginning of the MARG project, an important problem to be solved was the wireless communication form factor to be applied between the human user of the sensors and the server computer. Commercial-Off-The-Shelf wireless components that implement the 802.11 standards are available in different form factors, ranging from PCMCIA cards, USB modules, RS232 modules,

and multi-purpose modules. As a result, various solutions were examined. The most important of these potential form factors are discussed, with the reasons they were discarded.

### **1. The Use of a Wireless PCMCIA Card**

In the beginning of the MARG project, one of the ideas considered was to have a small computer (wearable computer) strapped onto the human carrying the sensors. The purpose of this computer would be to process the streaming information from the sensors. Then, the information would be transmitted to the server computer. With a wearable computer, one approach to the wireless transmission of the data would be the use of an 802.11b PCMCIA card.

Once this idea was closely examined, a number of problems surfaced. The wearable computer would have to be large enough to host an operating system that would support drivers for such a PCMCIA card. That approach was more or less out of the question; the reason for discarding it (besides the high cost) was that the computer had to be kept small and had to consume little power.

### **2. The Use of a Wireless USB Device**

Another idea that was examined was the use of the wireless 802.11b modules with a Universal Serial Bus (USB) interface. Since there are a variety of USB-based wireless communication devices, it seemed hopeful to examine the idea. One of the first devices considered was the NETGEAR 802.11b wireless USB network adapter, a small, simple and reliable device as shown in Figure 4.



**Figure 4. The NETGEAR Wireless USB Network Adapter [From Ref. 17].**

However, the USB module is designed for use on computers with a standard OS, such as Windows, and requires a driver. The MARG system uses a microcontroller (TI MSP430F149) that does not support a standard OS. Therefore, for the same reasons as the wireless PCMCIA card this idea was also discarded.

### **3. The Use of a RS232 Wireless Communication Device**

After the candidate smart devices were rejected, efforts were directed toward something more basic. A simple and very common interface among devices was considered, the RS232 serial interface, which does not require an OS. The use of an RS232 serial port was a very attractive idea, since it is a protocol that is easily implemented. Any computer or microprocessor can afford a serial output. After some research on existing wireless communication devices supporting a serial RS232 input, the 802.11b compatible OTC *WiSER* wireless communication device was located among other solutions. In an early implementation [Ref. 1], the *WiSER* module, shown in Figure 5, was used for wireless data transmission of three MARG sensors.



**Figure 5. The OTC WiSER Wireless Serial RS232 Network Adapter [From Ref. 18].**

However, the data rate of the *WiSER* module is limited to 115,200 bps, as is generally the case with all the commercial implementations of the RS232 protocol. Hence, it cannot be used for the MARG system with fifteen sensors (as mentioned earlier) mainly due to the following drawbacks:

- Its slowness (buffering issues),
- Its high-power consumption,
- Its size,
- The difficulty of integrating it into the CIU design.

#### **4. The Use of a Multi-purpose Wireless Communication Device**

Although a RS232 wireless communication device presents a solution, it is far from being the perfect one, meaning the search never ceased for a faster yet simpler communication protocol than the RS232 and a wireless communication module to support it.

The first encounter with the DPAC *airborne* wireless communication device was through the company's (DPAC) website. The idea of the DPAC *airborne*, as it was presented in the website, looked quite promising. The DPAC *airborne* is a multi-purpose module that supports, among others, the RS232, UART, and SPI. In addition, it does not require an OS. Its maximum data rate is 460,800 bps using UART and 20 Mbps using SPI. It is small in size (38 x 27 x 4.2 mm) and uses less power (drawing 200 – 420 mA of current) [Ref. 7]. After contacting the specific company and doing some research, the DPAC *airborne* wireless LAN node module evaluation and development kit was ordered, in order to be evaluated and tested. The goal was to prove that the DPAC *airborne* was capable of everything that the company claimed. Figure 6 shows a picture of the DPAC WLAN module.



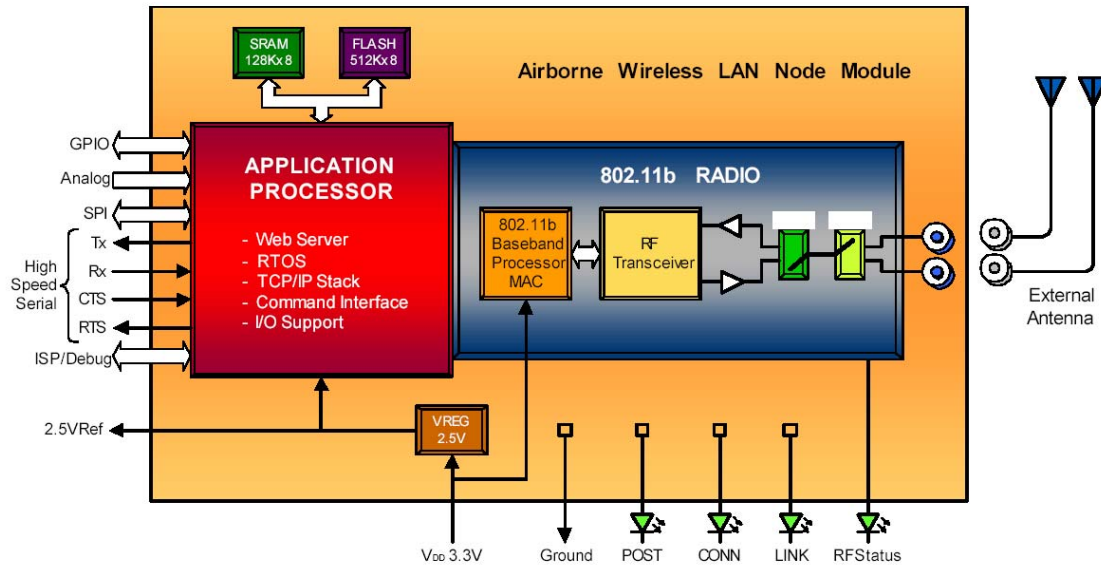
**Figure 6. The DPAC WLAN Module [From Ref. 19].**

The DPAC *airborne* represented the best solution available at the time and was therefore chosen for integration with the MARG system. More details about the DPAC *airborne* are provided below.

#### **D. THE DPAC *AIRBORNE* WIRELESS COMMUNICATION UNIT**

The DPAC *airborne* is a highly integrated 802.11b module [Ref. 7] fulfilling the performance requirements for the MARG project. Figure 7 is a hardware block diagram of the module. Its main features include [Ref. 7]:

- IEEE 802.11b wireless module,
- Build-in web-server, which enables the change of the set-up parameters in real time,
- Configurable serial, digital and I/O ports.



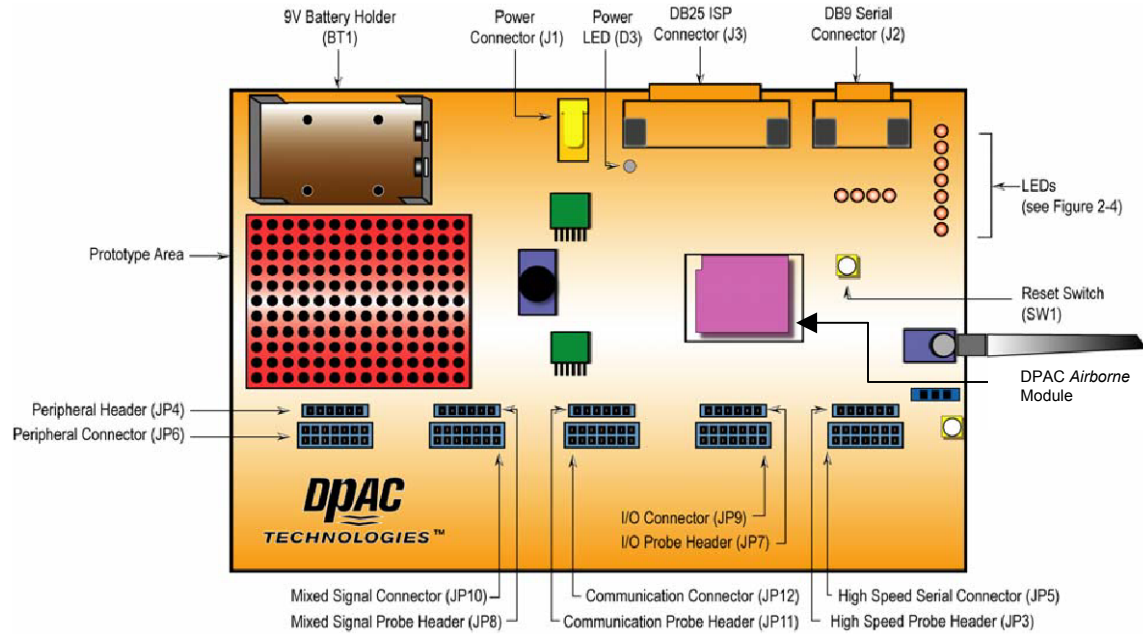
**Figure 7. The DPAC Airborne Hardware Block Diagram [From Ref. 7].**

Specifically, the DPAC *airborne* has among features a serial RS232 input-output port and a high-speed serial UART input-output port. The first was to be used in setting the module up and testing. The second was to be used as the high-speed communication channel needed between the TI microcontroller and the DPAC *airborne*.

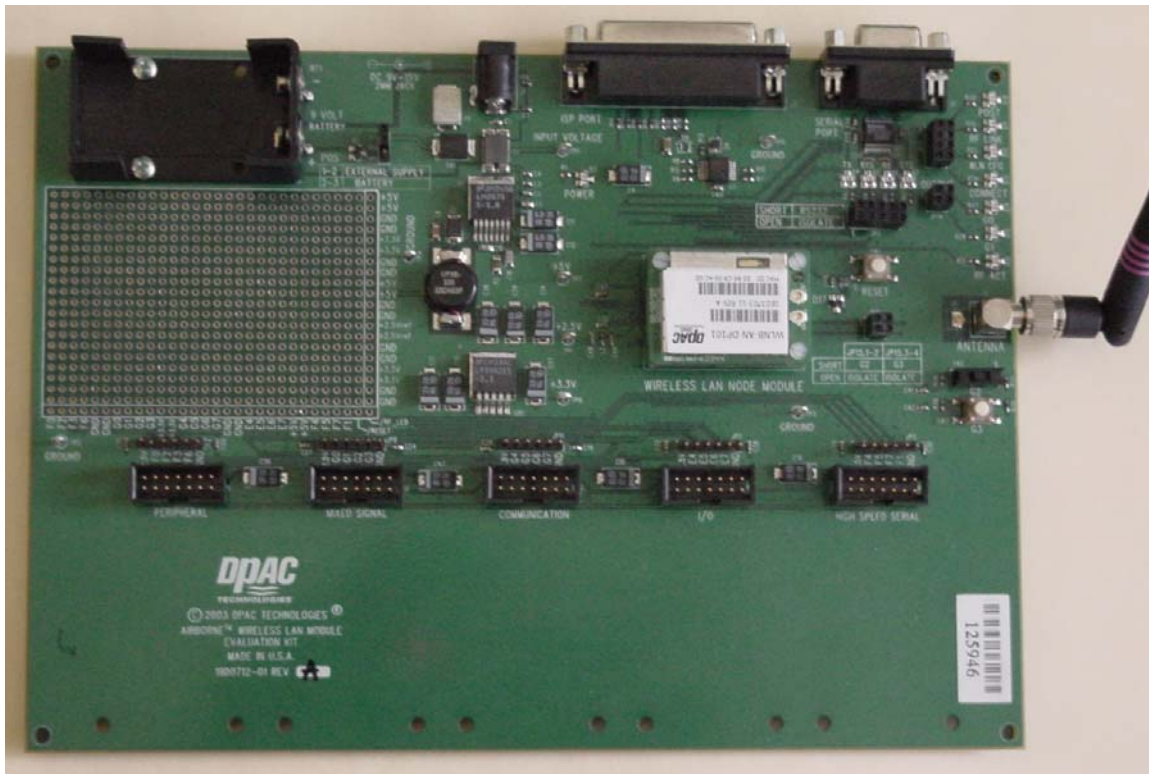
The speeds (data rates) supported by the high speed UART are 300, 600, 1,200, 2,400, 4,800, 9,600, 14,400, 19,200, 28,800, 38,400, 57,600, 115,200, 230,400 and 460,800 bps.

#### **E. THE DPAC AIRBORNE WLAN NODE MODULE EVALUATION AND DEVELOPMENT KIT**

The DPAC *airborne* came with a LAN module evaluation and development kit for testing and development purposes [Ref. 20]. The DPAC *airborne* evaluation board layout is shown in Figure 8, and Figure 9 is a photograph of the actual board.



**Figure 8. The DPAC Airborne Evaluation Board Layout [After Ref. 21].**



**Figure 9. The DPAC Airborne Evaluation Board.**

The evaluation board (EVB) has an easy-to-use layout and can serve as a programming socket for the DPAC modules, since it is quite easy to remove one and attach another one.

## **F. SUMMARY**

In this chapter, the wireless communication requirements of the MARG system were presented. After a brief description of the 802.11 wireless standards, the COTS components that could possibly be integrated into the MARG system were discussed. Finally, the DPAC *airborne* wireless communication module was introduced and presented.

The next chapter describes the step-by-step testing and evaluation conducted on the DPAC *airborne*.

THIS PAGE INTENTIONALLY LEFT BLANK

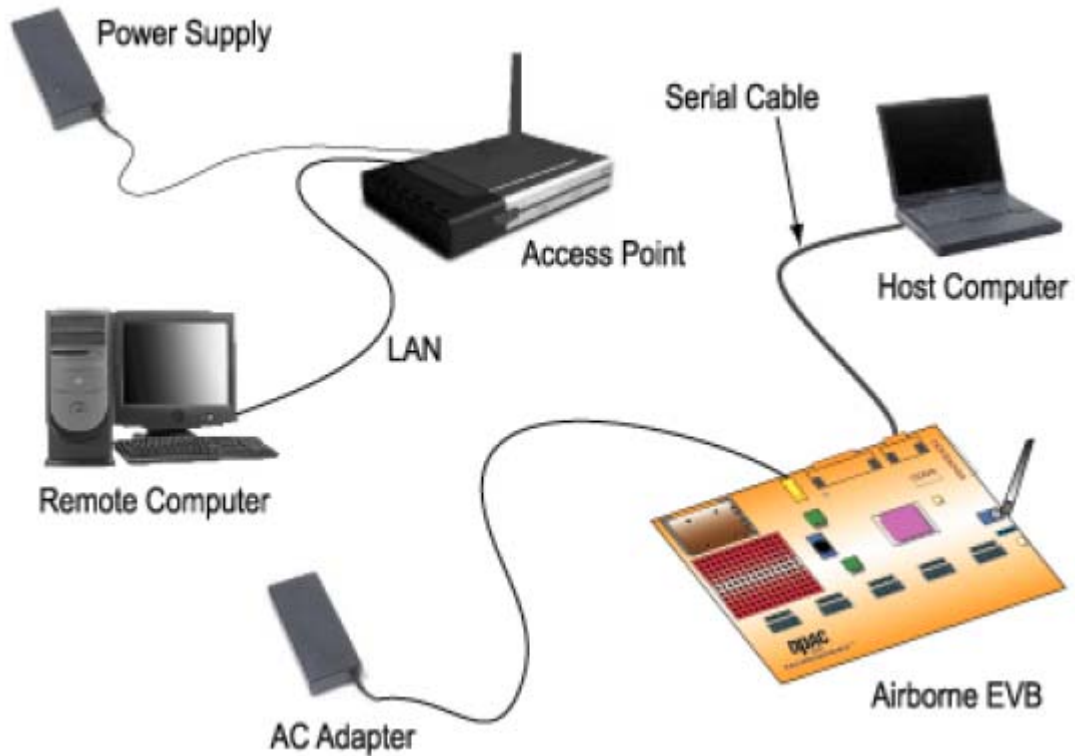
## IV. TESTING AND EVALUATION

The TI microcontroller was described in Chapter II. The DPAC *airborne* wireless communication module was discussed in Chapter III. In this chapter, the process of testing and evaluating the communication interface between the TI microcontroller and the DPAC *airborne* are presented. The whole process was divided into four parts. The first part is the procedure to initially communicate with the DPAC *airborne* and set it up. The second part is the procedure to establish communication between the DPAC *airborne* and a TCP server. The third part is the procedure to establish communication between the TI microcontroller and the DPAC *airborne*. The fourth part is the procedure to increase the baud-rate to the desired value of 230,400 bps.

### A. COMMUNICATING WITH THE DPAC AIRBORNE

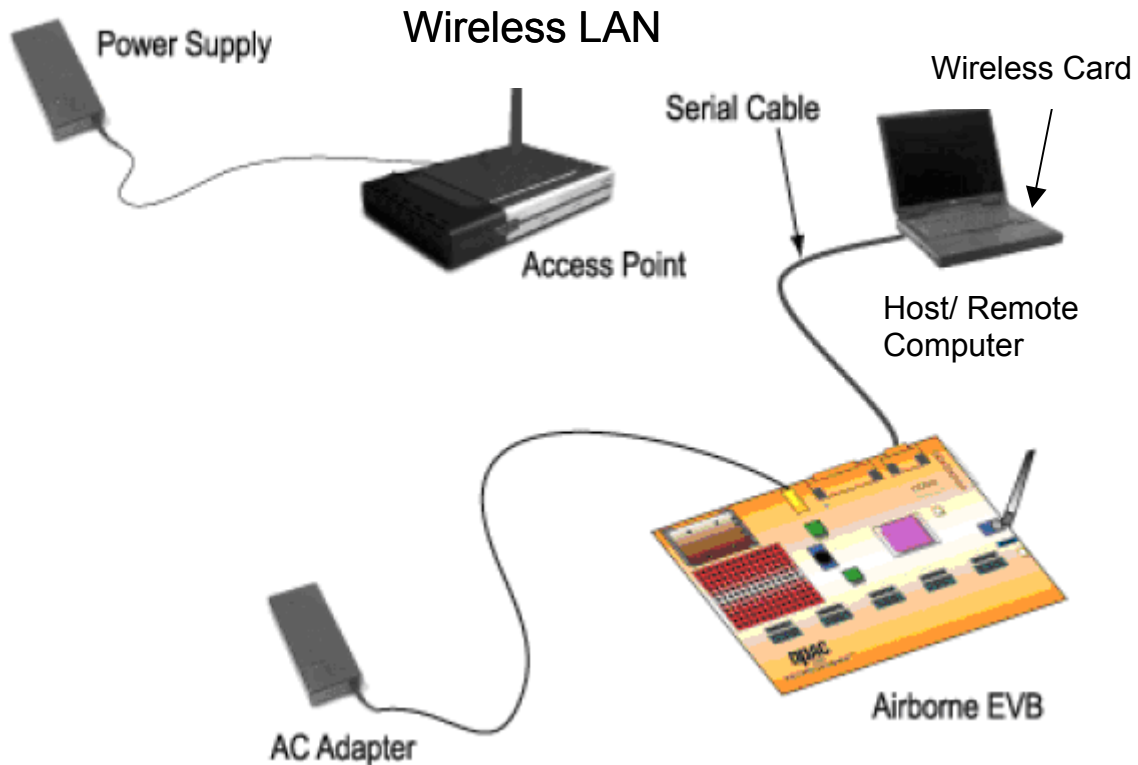
In order to set up the DPAC *airborne* the hardware configuration shown in Figure 10 is recommended in the user manual [Ref. 20].

The “Remote Computer” is the machine that receives data wirelessly from the DPAC *airborne* through the “Access Point.” The “Host Computer” is the device connected to the DPAC *airborne* serially, and for this thesis it is substituted by the single-board computer in the CIU, the TI microcontroller; and the serial connection is substituted by an UART connection.



**Figure 10. Hardware Configuration [From Ref. 20].**

Of course, the “Remote Computer” and the “Host Computer” can be on the same machine. In addition, the LAN cable shown can be substituted by a WLAN. For the purposes of this thesis, the hardware configuration shown in Figure 11 is adopted.



**Figure 11. The Hardware Configuration Used [After Ref. 20].**

On the “Host/Remote Computer” of this configuration, one application plays the role of the “Host Computer” connected serially to the DPAC *airborne* and another application, the “Remote Computer,” connected wirelessly (802.11b) to the DPAC *airborne*.

### **1. Setting up the DPAC Airborne**

In order to initialize the DPAC *airborne*, the following steps, as described in the “Quick Start Guide” [Ref. 20], must be followed:

- Initiate the wireless access point (AP) and set it up as a DHCP server so that it assigns an IP address to the DPAC *airborne* and “Remote Computer” automatically.

- Make sure that the “Remote Computer” is assigned an IP address by the DHCP server on the AP. This is accomplished by using the *ipconfig /release* and *ipconfig /renew* “command prompt” commands as shown in Figure 12.

```

C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Ioannis.NPGS-FSP45040KU>ipconfig /release

Windows IP Configuration

Ethernet adapter Wireless Network Connection 2:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 0.0.0.0
    Subnet Mask . . . . .             : 0.0.0.0
    Default Gateway . . . . .         : 

C:\Documents and Settings\Ioannis.NPGS-FSP45040KU>ipconfig /renew

Windows IP Configuration

Ethernet adapter Wireless Network Connection 2:

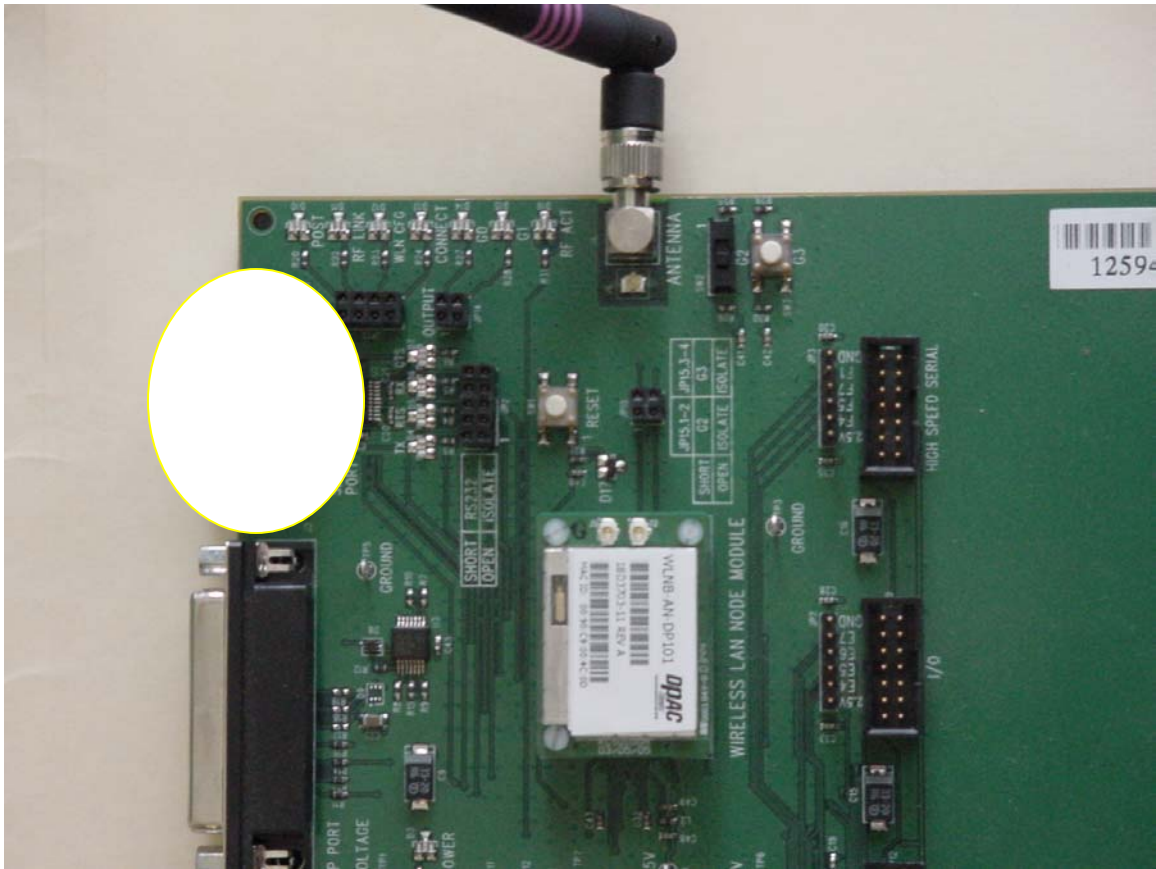
    Connection-specific DNS Suffix  . : domain
    IP Address. . . . .               : 192.168.1.24
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 192.168.1.1

C:\Documents and Settings\Ioannis.NPGS-FSP45040KU>

```

**Figure 12. Verifying the “Remote Computer’s” IP Address.**

- Use the supplied serial cable to connect the RS232 serial port connector on the DPAC *airborne* EVB (shown in Figure 13) to the “Host Computer” [Ref. 20].



**Figure 13. The Serial Port Connector on the DPAC Airborne EVB.**

- After the connections are made, the next step is to start a terminal-emulation program (e.g., HyperTerminal) on the “Host Computer.” The emulation program must initially be configured to meet the following parameter values:

Bits per Sec	Data Bits	Parity	Stop Bits	Flow Control
9,600	8	None	1	None

**Table 1. HyperTerminal Configuration for the Initial DPAC Airborne EVB Set-up.**

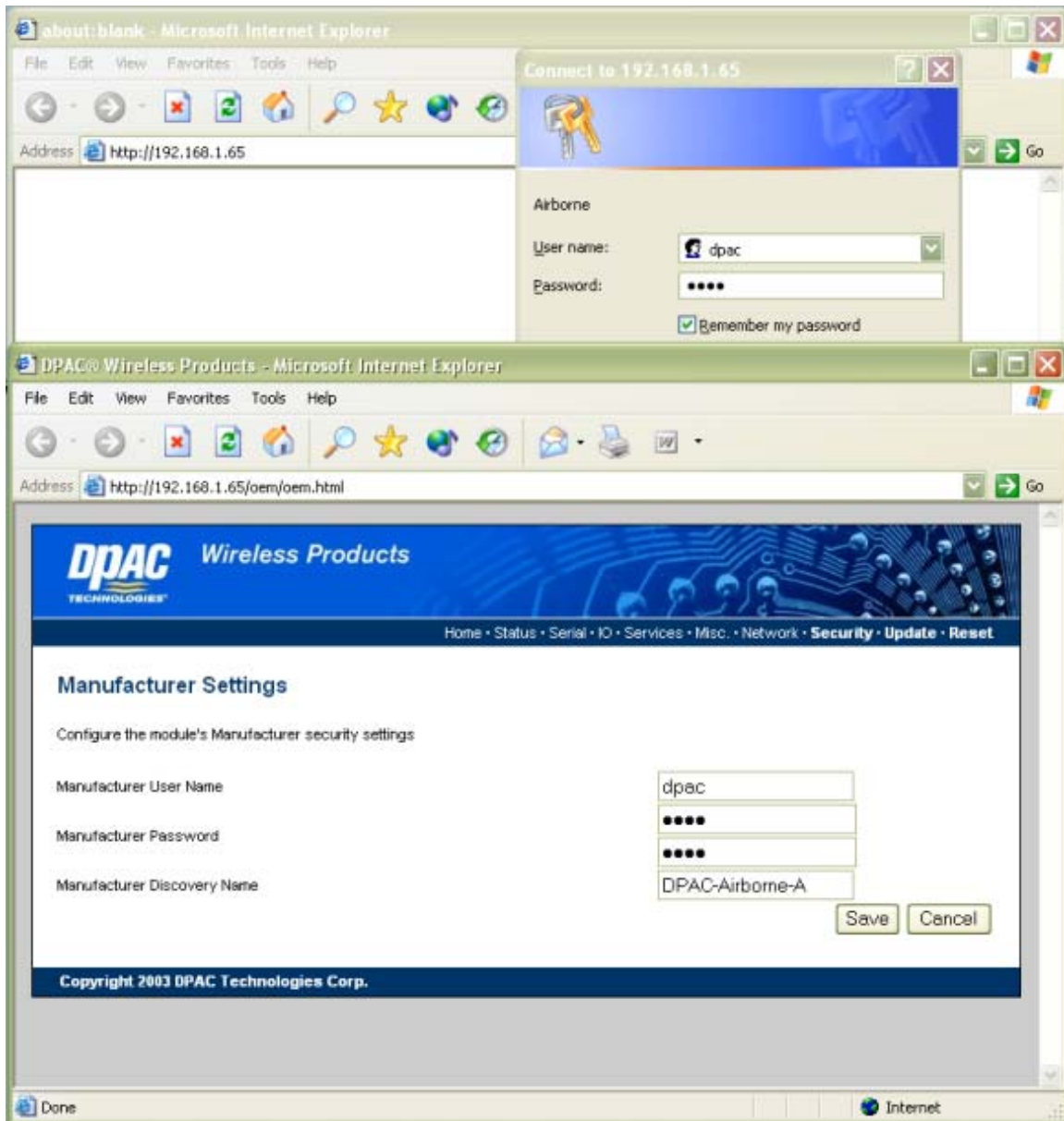
- In the next step of the initialization of the DPAC *airborne*, command line interface (CLI) commands are used in the communication between the emulation program and the wireless communication device. The two following steps must be executed: (1) logging in, and (2) finding and saving the service set identifier (SSID) of the wireless AP that is to be used. At this point, the DPAC *airborne* can communicate with the wireless AP. From the first communication of the DPAC *airborne* with the AP, the DPAC *airborne* is assigned an IP address, which can be retrieved through the CLI on the emulator screen.

## **2. Using the Web Browser**

After the initial set-up and the designation of an IP address to the DPAC *airborne* wireless communication module, the DPAC *airborne* can now easily be accessed using a common web browser. Instances of the connection procedure using Internet Explorer are shown in Figure 14.

Through the web browser the following parameters, among others, can be set or changed [Ref. 7]:

- User names and passwords.
- The serial port settings: baud rate, data bits, parity, flow control.
- The network connection settings: LAN server IP address, LAN server port, session inactivity timeout (in seconds), the default host mode and the connectivity retry time.



**Figure 14. Connecting to the DPAC Airborne Using Internet Explorer.**

In addition to setting various parameters, one can view the working status of the module including link status, port status, service set identifier (SSID), media access control address (MAC address), transmit rate, communication quality, signal level, noise level, own IP address and default gateway. Browser instances depicting the above information are shown in Figure 15.

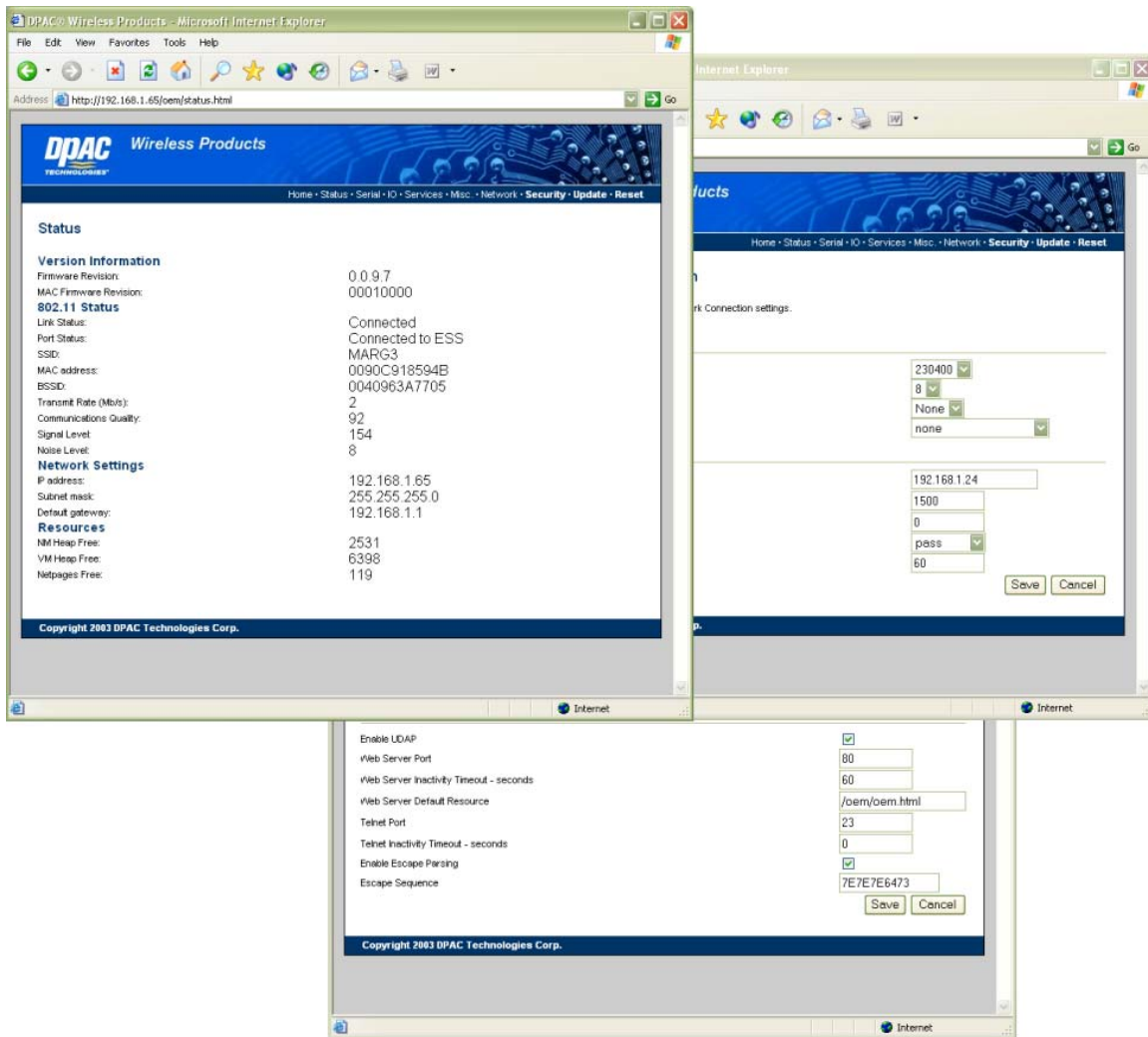


Figure 15. Web Browser Instances.

## B. ESTABLISHING COMMUNICATION BETWEEN THE AIRBORNE WIRELESS COMMUNICATION MODULE AND A TCP SERVER

Establishing communication between the DPAC *airborne* and a TCP server involves among other features having a TCP server program (which must be created) run on the “Remote Computer.” This server should be able to print all incoming data (from the client/*airborne*) on the screen and print the input from the keyboard on the screen before sending it to the client. In addition, the IP address of the “Remote Computer” and port number where the TCP server is listening must be known and set to the *airborne*.

In this first testing configuration a very low data rate was used, namely 2,400 bps, since the goal was to just establish communication.

The HyperTerminal used in the connection between the “Host Computer” and the *airborne* (through the RS232 serial port) acts as the *airborne*’s screen/monitor.

In order to establish communication, a connection between the “Host Computer” and the *airborne* must be initialized and also a wireless connection between the *airborne* and the server on the “Remote Computer” must be established. In other words the following communication path must be followed/established: hyper terminal – serial RS232 – *airborne* – wireless 802.11b – TCP server.

The final result is that when typing a character into the HyperTerminal (“Host Computer”) this character is transferred through the serial connection (RS232) to the *airborne*. Then from there (TCP client), it is transmitted wirelessly (802.11b) to the TCP server (“Remote Computer”) and printed on the screen. This sequence can be repeated in the reverse direction also.

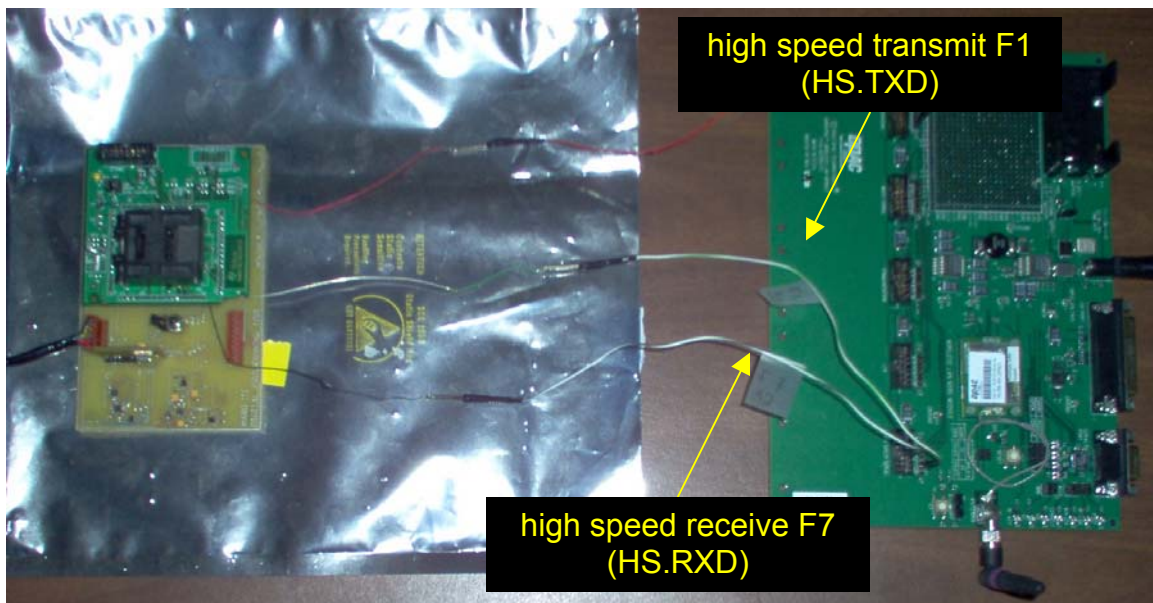
### **C. ESTABLISHING COMMUNICATION BETWEEN THE *AIRBORNE* WIRELESS COMMUNICATION UNIT AND THE TI MICROCONTROLLER**

After verifying that the *airborne* was communicating correctly through both the serial port and the wireless antenna, the next step was to establish the following communication: TI microcontroller ↔ UART ↔ *airborne* ↔ wireless 802.11b ↔ TCP server, still under a low data rate.

Here the serial connection to the “Host Computer” was substituted by a UART connection to the TI microcontroller. This was accomplished by leading the data from the DPAC *airborne* to the UART port instead of the RS232 serial port. This function was accomplished by removing all the jumpers from the jumper group JP2 [Ref. 21].

From the UART port the pins F1 (transmit) and F7 (receive) were selected and connected to their counterparts on the TI microcontroller.

At this point, of the entire communication configuration that must be tested (TI microcontroller ↔ UART ↔ airborne ↔ wireless 802.11b ↔ TCP server) only the part between the TI microcontroller and the DPAC *airborne* has not yet been verified. Figure 16 shows how the TI microcontroller was connected to the DPAC *airborne* evaluation board.



**Figure 16. The TI Microcontroller – DPAC Airborne Connection.**

In order to complete the verification process, the next step is to program the TI microcontroller to respond to incoming UART signals in such a way that tests can be run. An echo program was used that allows the TI microcontroller to echo back every piece of data it receives.

The result of all the above (after some trial and error) was that, when typing a character into the server window (“Remote Computer”), this character was transferred through the wireless connection (802.11b) to the *airborne*, and then from there transmitted through the UART to the TI microcontroller. The TI micro-

controller “echoed” the character and following the same path this character was printed on the screen of the “Remote Computer.”

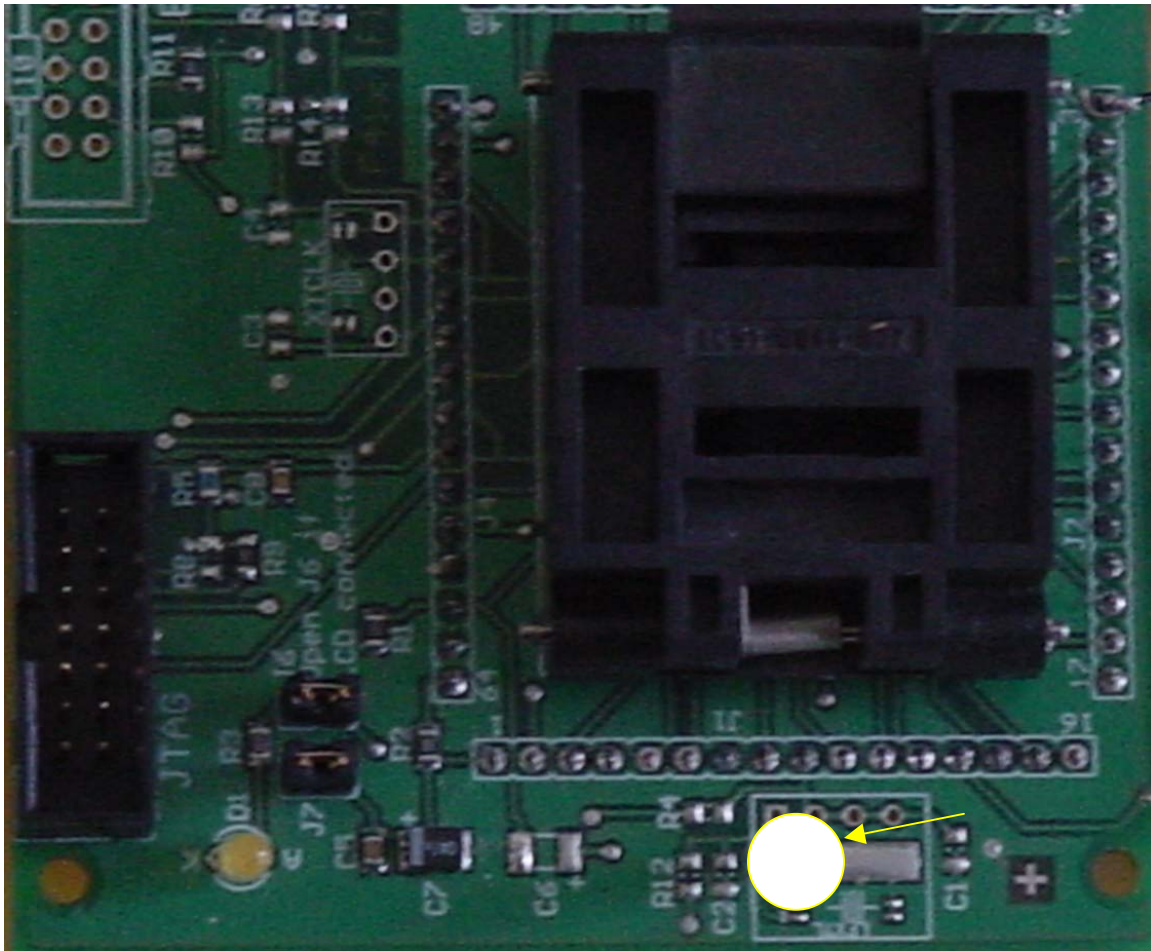
#### **D. INCREASING THE BAUD-RATE**

Establishing the above communication pattern was certainly a success but the data rate (2,400 bps) was too low. The DPAC *airborne* has the attribute of being able to increase the data rate to high levels with no difficulty, yet a way had to be found to do the same with the TI microcontroller.

##### **1. Using the Stock Crystal (32,768 Hz)**

As mentioned earlier in Chapter II, the correct timing for the UART mode is sustained by a clock/crystal, which also provides the selected data rate. Furthermore, on the TI microcontroller, the clock/crystal used to sustain that data rate can be selected among the available clocks of the microcontroller. For the purposes of this thesis, the auxiliary clock driven by a common external crystal was used. The position of the external crystal is indicated in Figure 17.

The goal at this point is to establish the above-mentioned communication pattern generating the maximum data rate allowable by the stock crystal of the TI.



**Figure 17. The External Crystal of the TI Microcontroller.**

The data rate (baud-rate) generator on the TI microcontroller is capable of generating conventional data rates from almost any (some restrictions apply) crystal/ source frequency [Ref. 9], i.e., even from frequencies that when divided by the desired data rate do not yield an integer. The data rate generator first employs a simple divider, which divides the source frequency by the integer part of the frequency – data rate ratio. Then a modulation data shift register is employed in order to compensate for the difference between the integer divider and the actual result of the division.

The modulation data shift register is composed of a number of bits that indicate the time delay required for the data rate to be accurate. For the purpose of

the generation of these modulation bits, a MATLAB program was created [Appendix A]. The MATLAB program was tested with the various given pairs of the data rate – modulation-bits (for specific crystal frequencies).

## **2. Using an Aftermarket Crystal (7,372,800 Hz)**

In order to increase the data rate to the required 230,400 bps, a crystal of 7,372,800 Hz replaced the stock external crystal (32,768 Hz) and the assembly program driving the TI microcontroller was changed correspondingly.

This particular crystal frequency was chosen for the following reasons:

- The crystal frequency divided by the desired data rate returns an integer. In that way, the produced data rate will be more accurate/exact than if the result was not an integer (no modulation bits needed in this case) [Ref. 9],
- This integer is greater than thirty (32), and in that way the probability of error in the transmitted data because of the inaccurate data rate is minimized (going to a higher ratio than that does not enhance the probability of error much) [Ref. 9],
- In addition, this crystal frequency is widely available from several crystal vendors.

After the replacement of the crystal, the communication data rate between the TI microcontroller and the DPAC *airborne* actually was increased to 230,400 bps. This was confirmed by using the testing sequence described in the previous section, except that the data-rate setting at both ends was set as 230,400 bps instead of 2,400 bps.

## **E. SUMMARY**

This chapter presented the testing and evaluating of the communication interface between the TI microcontroller and the DPAC *airborne*. The conclusion of this chapter is that the thesis goals are feasible and the next chapter presents how they are implemented.

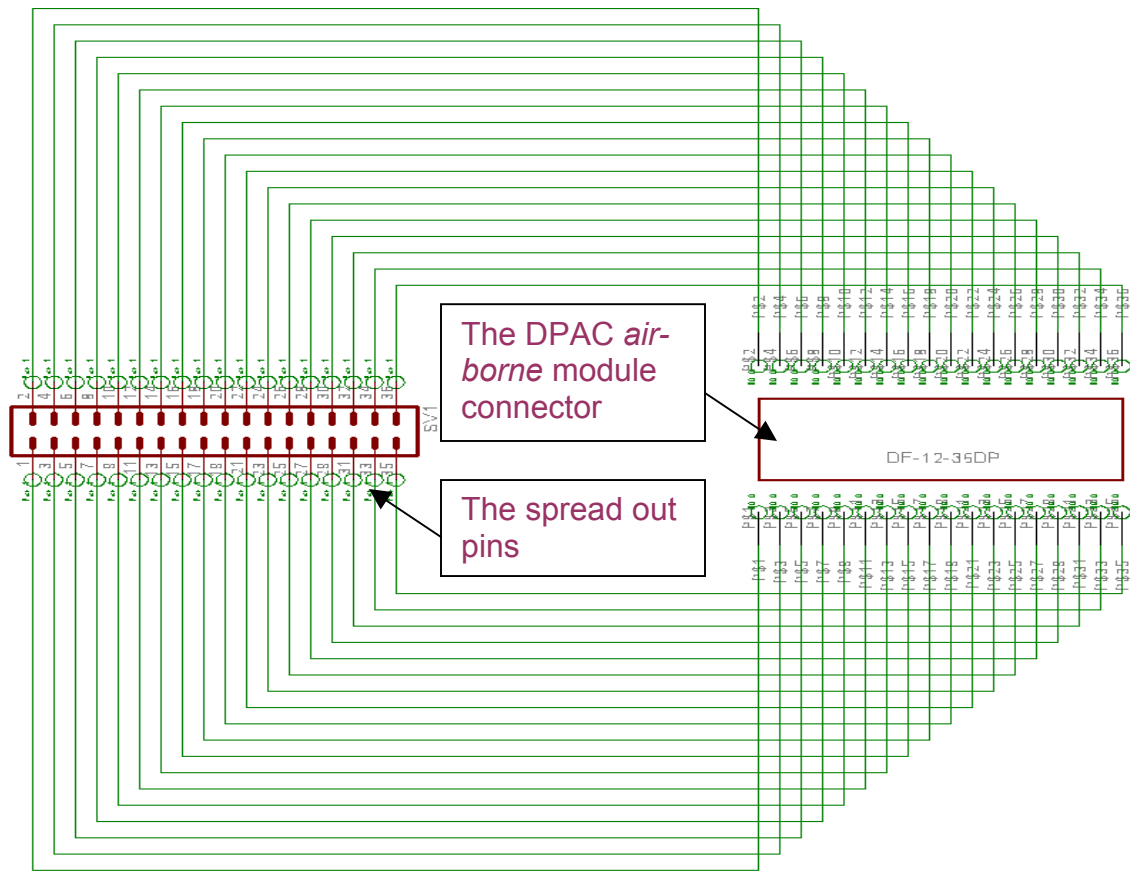
## **V. LINKING THE TI MSP430F149 AND THE DPAC *AIRBORNE* COMMUNICATION UNIT**

This chapter describes the design and fabrication of the DPAC *airborne* module board and it describes the testing performed. In addition, it presents how the TI microcontroller was connected to the DPAC module board and the initiation of their communication.

### **A. THE DESIGN OF THE DPAC *AIRBORNE* MODULE BOARD**

The DPAC *airborne* wireless communication module evaluation board (EVB) was very helpful in testing and verifying the performance of the WLAN module. However, its functionality is more than what is needed on this project and its size is too big. Therefore, a much simpler and smaller board had to be designed. The purpose of this board was to prove that it is possible to integrate the DPAC *airborne* into the CIU.

The design utility used was the freeware version of the EAGLE software (Easily Applicable Graphical Layout Editor, Version 4.11 for Windows, Light Edition) from CadSoft [Ref. 22]. First, a schematic was drawn, shown in Figure 18.

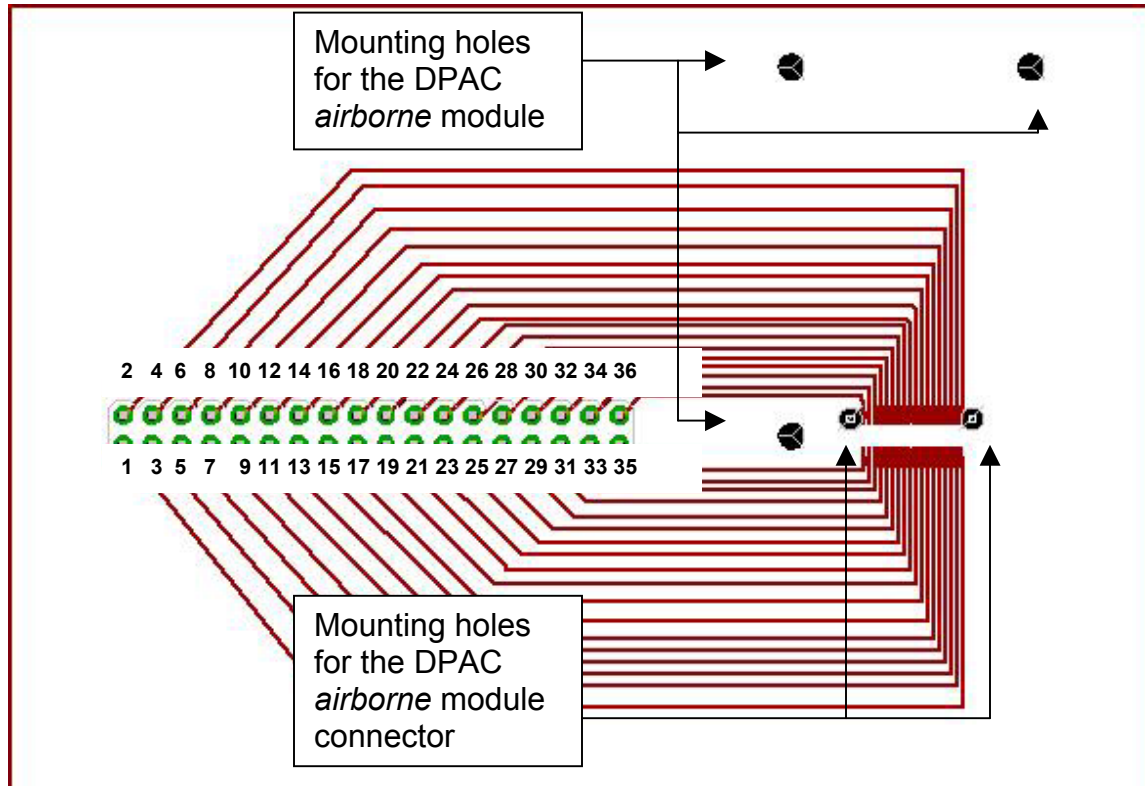


**Figure 18. The Schematic of the DPAC *Airborne* Module Board Layout (not to scale).**

The schematic is just a spread-out of the pins of the module. This way, the board layout can be first evaluated and confirmed as to what connections are needed for a standalone module. When drawing the schematic, the correct spacing had to be calculated for the pins of the DPAC module connector (from [Ref. 7]) and then those pins were connected to a standard connector (left side of the schematic). On this standard connector, the spacing is adequate so that all the pins can be used and interconnected.

Second, the schematic was converted into the board layout seen in Figure 19, where the position of the mounting holes for the DPAC *airborne* module and

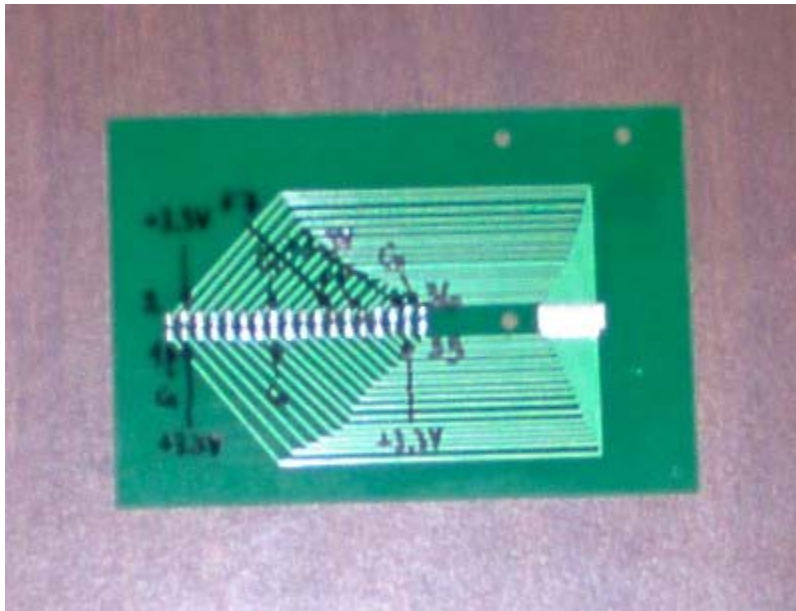
the DPAC *airborne* module connector can be seen as they were calculated from [Ref. 7]. This concluded the first stage of designing the board layout. The next stage is the fabrication.



**Figure 19. The DPAC Airborne Module Board Layout.**

## **B. THE FABRICATION OF THE DPAC AIRBORNE MODULE BOARD**

The laboratories at NPS do not have the capability of fabricating this type of circuit boards. After inquiring about companies fabricating circuit boards, it was decided to contract the fabrication to PCB FAB EXPRESS of Sunnyvale, California. PCB FAB EXPRESS fabricated five boards for \$260. The schematic and layout of the board had to be converted into a number of different files (\*.cmp, \*.drl, \*.erc, \*.gpi, \*.plc, \*.pro, \*.sol, \*.stc, \*.sts, \*.whl) containing all the information about the board in a format compatible with the board/circuit printers used by PCB FAB EXPRESS. A few days after submitting the order, the board seen in Figure 20 was received.



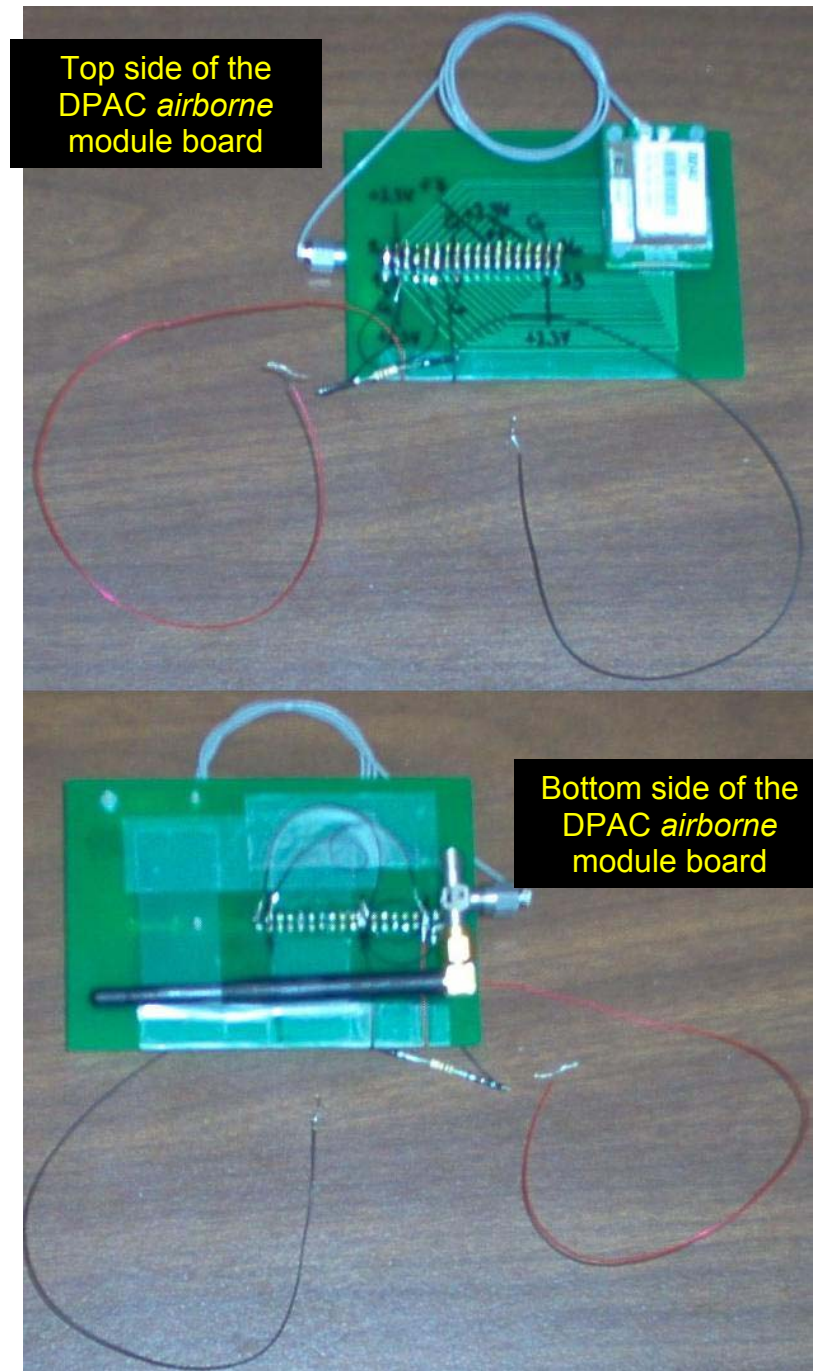
**Figure 20. The DPAC *Airborne* Module Board.**

### **C. THE EVALUATION AND TESTING OF THE DPAC *AIRBORNE* MODULE BOARD AND ITS CONNECTION WITH THE TI MICROCONTROLLER**

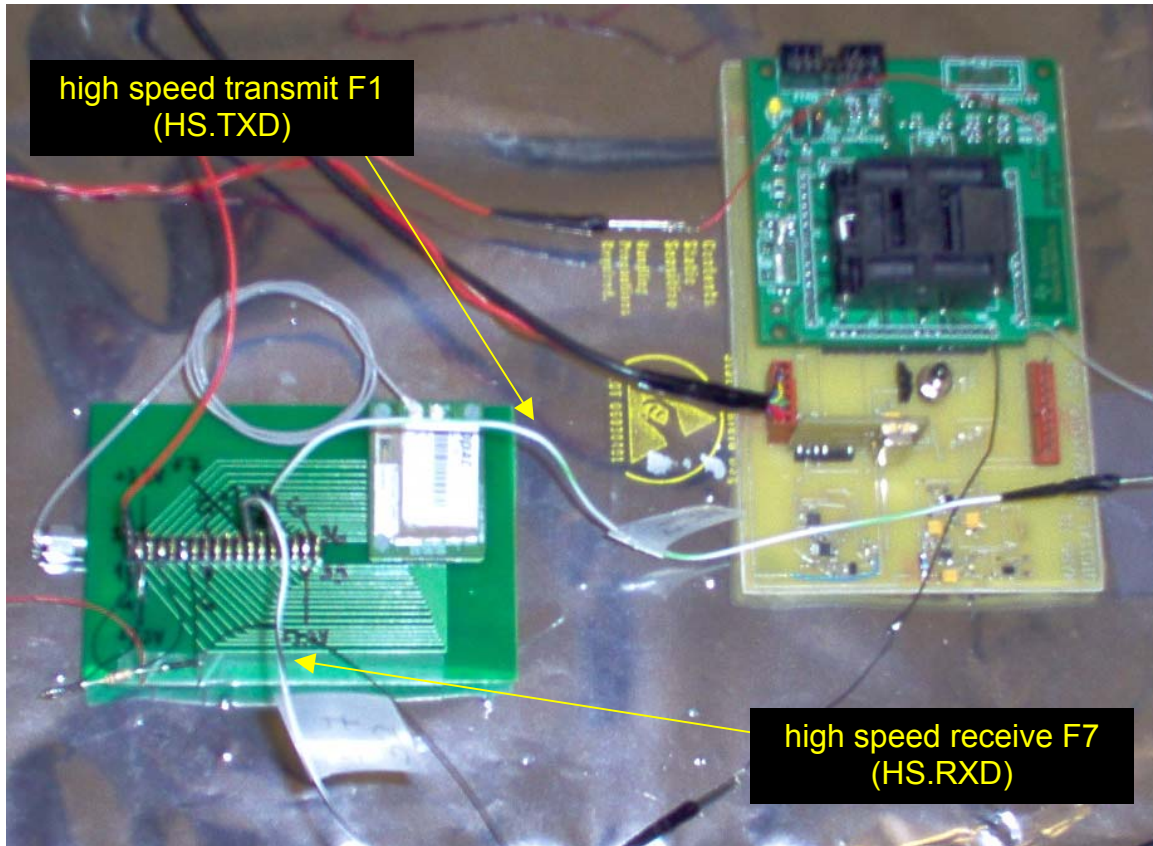
After a thorough study and testing, the modifications/additions that had to be made to the board were identified as the following [Ref. 7]:

- The +3.3 Volts power supply pins 3, 4, 33, and 34 had to be interconnected,
- The ground pins 1, 15, 16 and 36 had to be interconnected,
- The reset pin (7) had to be held high by being connected to a power pin, since a logical low initiates the reset sequence,
- Pin 11 had to be held high also (via a 10,000 Ohm resistor) in order to avoid the module to reset to factory defaults during boot,
- A 2.4 GHz antenna had to be connected to the module (802.11b standard).

After all the shortcomings were addressed and the modifications/additions were made, this board was connected to the TI microcontroller. The finished board is shown in Figure 21, and the board connected to the TI microcontroller in Figure 22.



**Figure 21. The DPAC Airborne Module Board.**

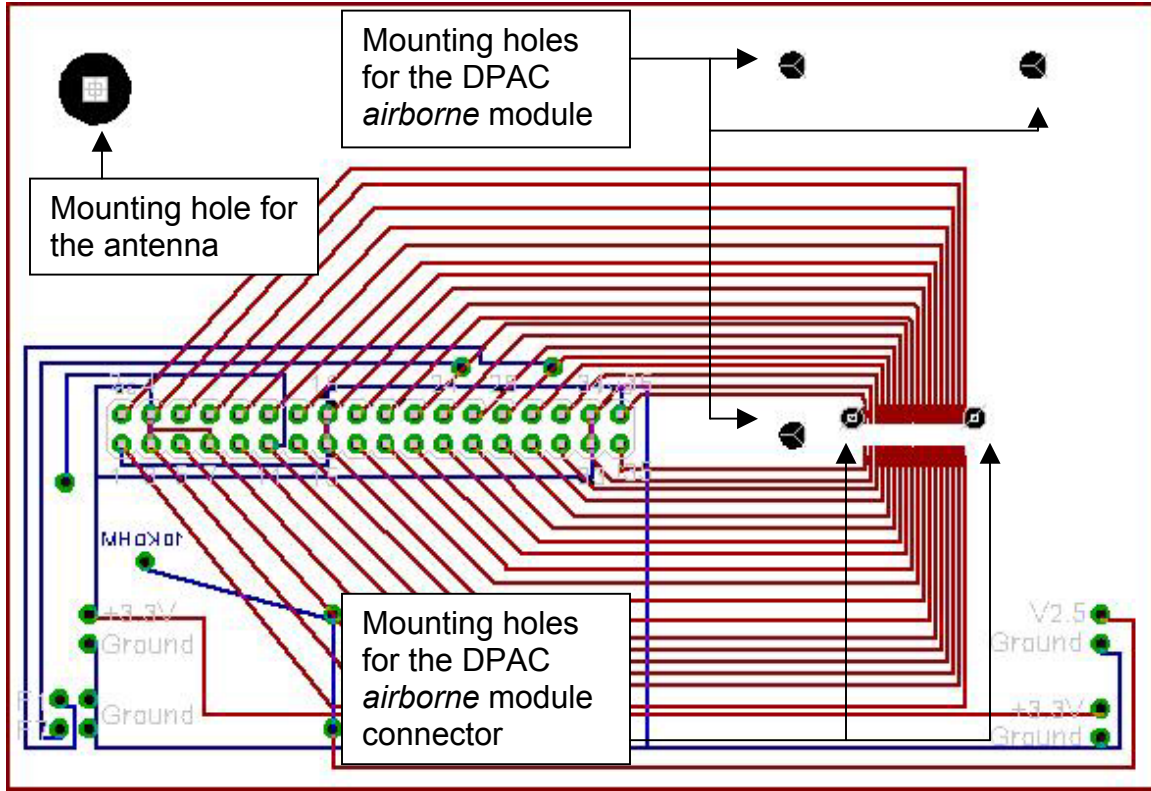


**Figure 22. The DPAC *Airborne* Module Board Linked to the TI Microcontroller of the CIU.**

In order to test the communication link, the testing sequence described in the previous chapter was repeated, only this time the fabricated DPAC *airborne* module board was used instead of the DPAC EVB.

A high-speed communication channel between the TI microcontroller and the DPAC *airborne* module was achieved. Therefore, the communication testing confirmed the functionality of the DPAC *airborne* module board and in general the feasibility of the DPAC *airborne* idea. The next step would be to integrate the DPAC *airborne* module into the design of the CIU. In order to achieve that, a de-

sign had to be implemented using all the additions/modifications mentioned in the beginning of the section. This design would be something similar to the layout shown in Figure 23.



**Figure 23. The Optimized DPAC Airborne Module Board Layout.**

#### **D. SUMMARY**

In this chapter, the implementation and linking of the DPAC *airborne* module board with the TI microcontroller were described. During this process, the research leading to the final DPAC *airborne* module board was presented.

The following chapter presents the overall conclusions of this thesis and suggestions for further development.

THIS PAGE INTENTIONALLY LEFT BLANK

## VI. CONCLUSIONS – FURTHER DEVELOPMENT

This final chapter of the thesis presents conclusions and suggestions for further development and optimization of the MARG system.

### A. THESIS CONTRIBUTIONS

This thesis has evolved the initial idea of tracking the motion of a human body into a feasible state. By now it is reasonable to state that a full-body representation, using all fifteen necessary sensors (needed for a virtual environment representation), will be accomplished within the next few quarters.

The main components of the current MARG project system are the MARG sensors (obviously), the CUI, the DPAC *airborne* serial-to-wireless communication unit, and a TCP server for receiving the data and sending them to the application used to depict the motion. At this point, multiple clients can connect to the server (at any time) and observe the real-time representation of the motion [Ref. 23].

The heart of the present system as well as the main component studied in this thesis is the DPAC *airborne* WLAN module and its interconnection with the TI microcontroller of the CIU. At this point the WLAN module can support the real-time transmission needs (real-time relay from the CIU) of up to fifteen sensors, all the sensors needed to represent the motion of the human body. After the initial evaluation and testing phase, the DPAC *airborne* module was integrated into the design of a board that is custom-made to fit the needs of the project. The evaluation of DPAC was a multi-stage process. First, the WLAN module had to be tested to verify whether it complied with what was advertised. Second, a way had to be found to make the TI microcontroller of the CIU accelerate to a transmission rate of 230,400 bps. Then the two units had to be integrated to work together smoothly.

The DPAC *airborne* WLAN module, among other things, also adds a great deal of modularity to the MARG project. The way the main components were configured and connected, the DPAC *airborne* can be taken out at any time and effortlessly substituted.

## **B. SUGGESTIONS FOR FURTHER DEVELOPMENT**

For the MARG project to reach a completion stage, the only element missing is the evaluation testing of the fifteen-channel CIU.

Beyond that, and after the whole system is operational, the only modification that could be attempted is to go from the UART protocol to the Serial Peripheral Interface (SPI) protocol. The SPI protocol is a synchronous serial interface that can achieve data rates as high as 20 Mbps [Ref. 7]. The reasons to switch from UART to SPI are mainly the higher data rate (which could increase even more in the future), and the fact that it is a synchronous transmission method, which is more stable over time.

The SPI protocol is a simple hardware interface primarily used to transfer data. It is full duplex and uses four wires/signals. It is (in contrast to UART) a synchronous serial interface based on a Master/Slave relationship. The SPI signals are the following four (whereas UART has only two):

- Serial Clock (SCK),
- Slave Select (SS),
- Master Out Slave In (MOSI), and
- Master In Slave Out (MISO).

On the DPAC *airborne*, the transition from UART to SPI (slave) is straightforward. The MOSI and MISO signals use the same wires as the RXD and TXD signals of the UART, respectively. In addition, the SCK and SS signal would have

to be wired to the corresponding pins of the microcontroller. To work in SPI mode no software changes are necessary on the DPAC *airborne*, but only on the microcontroller.

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX: THE MATLAB PROGRAM FOR THE CALCULATION OF THE MODULATION DATA SHIFT REGISTER

The modulation data shift register is composed of eight bits and is meant to apply to the microcontroller the time delay required for the data rate to be accurate.

When the frequency of the source clock/crystal is an exact multiple of the desired data rate, no modulation data shift register is required since the desired data rate can be generated just by dividing the clock frequency. Therefore, the modulation data shift register bits are set equal to zero.

When the frequency of the source clock is not an exact multiple of the desired data rate, then the bits that compose the modulation data shift register indicate the time delay required for the data rate to be accurate and must be calculated.

The following MATLAB program calculates the eight modulation data shift register bits, using the approach of [Ref. 6], and prints them on the screen, in a 2x4 matrix. In addition, it calculates and prints out the theoretical percent timing error on the screen. The essential inputs to the program are the desired data rate, the crystal frequency and a matrix 256x8 (named “m8”), containing the numbers from zero to 255 in eight-bit binary format.

```
% Ioannis R. Saliaris  
% March 2004  
% The MATLAB program for the calculation  
% of the modulation data shift register
```

```
clc  
format short
```

```
% desired data rate  
datarate=9600;
```

```
% source clock/crystal frequency  
BRCLK=32768;
```

```

UxBR=(BRCLK-mod(BRCLK, datarate))/ datarate;

mi=[ones(256,1),m8,[0 1 1 0 1 0 0 1 1 0 0 1 0 ...
    1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 ...
    1 0 0 1 1 0 0 1 0 1 1 0 ...
    0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 0 1 0 ...
    1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 1 0 ...
    0 1 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 ...
    1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 ...
    1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 ...
    0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 ...
    0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 ...
    1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 ...
    1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0]';zeros(256,1)];

for i=1:256
    m=mi(i,:);
    mm=0;
    for j=0:9
        er(j+1)=( datarate /BRCLK)*((j+1)*UxBR+(mm+m(j+1)))-(j+1);
        mm=mm+m(j+1);
    end
    error(i)=max(abs(er))*100;
    er=[];
end

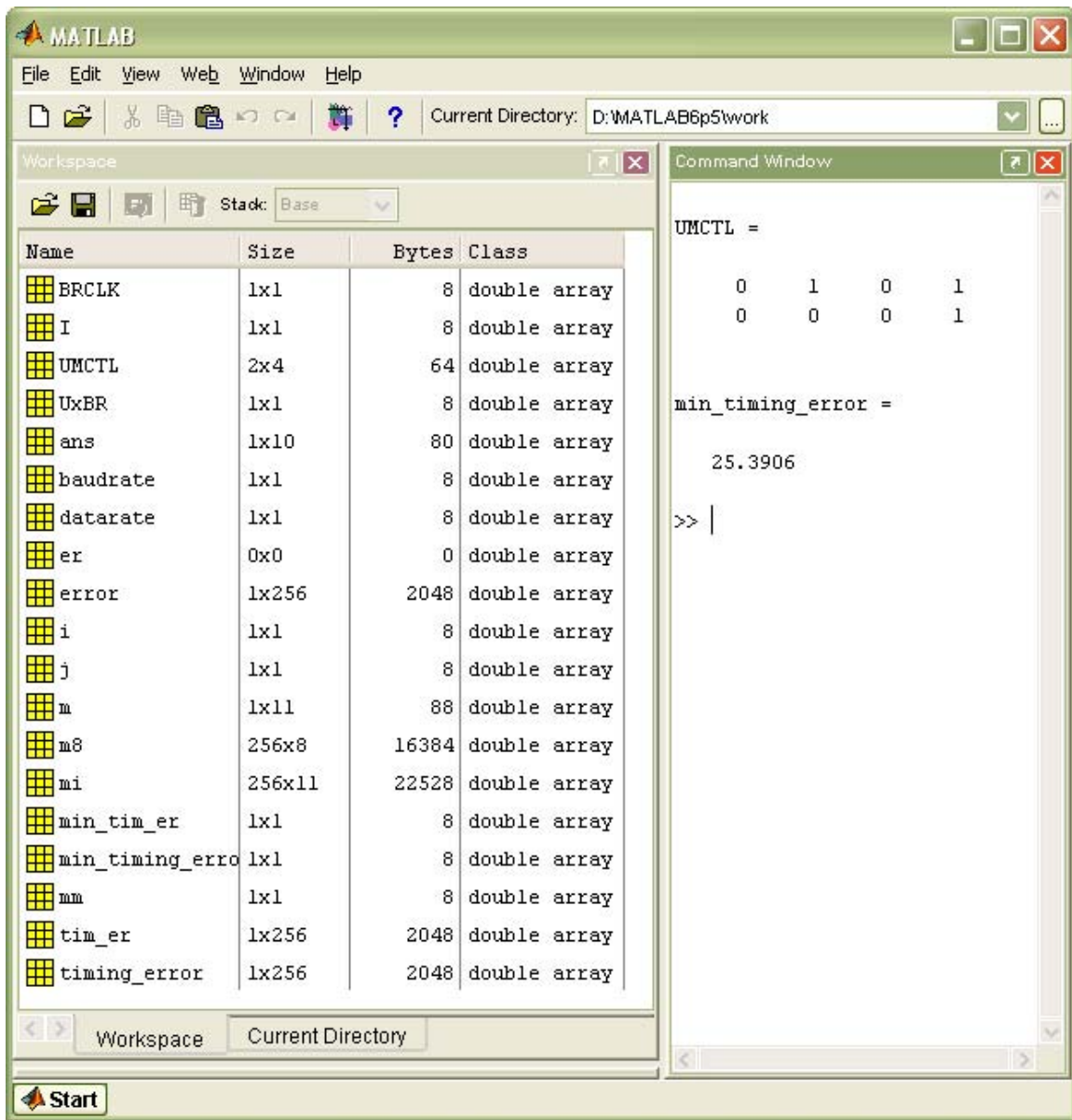
tim_er=error;
[min_tim_er,I]=min(error);

% the modulation data shift register bits
UMCTL=[mi(I,8),mi(I,7),mi(I,6),mi(I,5);mi(I,4),mi(I,3),mi(I,2),mi(I,1)]

% the theoretical percent timing error
min_timing_error=min_tim_er

```

Figure 24 shows how the command window and the workspace of MATLAB looks after the program is run.



**Figure 24. The MATLAB Window Screenshot Showing the Output of the Program.**

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF REFERENCES

1. Andreas Kavousanos-Kavousanakis "Design and Implementation of a DSP-Based Control Interface Unit (CIU)," MS Thesis, Naval Postgraduate School, Monterey CA, March 2004.
2. Synthetic Environment Laboratory, <http://www.sel.bee.qut.edu.au/understand/understand.htm>, last accessed January 2004.
3. Eric Bachmann, "Inertial and Magnetic Tracking of Limb Segment Orientation for Inserting Humans into Synthetic Environments," Ph.D. Dissertation, Naval Postgraduate School, Monterey CA, December 2000.
4. R. E. Smith, <http://www.rs485.com/rs485spec.html>, last accessed April 2004.
5. SIXNET, <http://www.sixnetio.com/htmlhelps/vtmodem/5a58b3e.htm>, last accessed April 2004.
6. MSP430F149 Product Information, Texas Instruments Incorporated, 1995-2003, <http://focus.ti.com/docs/prod/folders/print/msp430f149.html>, last accessed January 2004.
7. Airborne WLN Module Data Book 39L3702-01B, pdf file from DPAC Technologies Corporation, Garden Grove CA, 2003.
8. MSP430F149 image, Texas Instruments Incorporated, <http://www.ixbt.com/mobile/images/gm918/chip.gif>, last accessed February 2004.
9. Texas Instruments MSP430x13x, MSP430x14x, MSP430x14x1 Mixed Signal Microcontroller Datasheet, Texas Instruments Incorporated, July 2000, Rev E, August 2003, <http://focus.ti.com/lit/ds/symlink/msp430f149.pdf>, last accessed January 2004.
10. James Thornton, <http://jamesthornton.com/freebsd/articles/serial-uart/>, last accessed February 2004.
11. Hewlett Packard, [http://h18000.www1.hp.com/products/wireless/wpan/files/WhitePaper\\_BluetoothTechnologyOverview-QA.pdf](http://h18000.www1.hp.com/products/wireless/wpan/files/WhitePaper_BluetoothTechnologyOverview-QA.pdf), last accessed March 2004.
12. Antonios Varelas, "An Investigation of Wireless Solutions for the 'Last Mile,'" MS Thesis, Naval Postgraduate School, Monterey CA, March 2004.

13. ZDNet UK, <http://insight.zdnet.co.uk/communications/wireless/0,39020430,2132483,00.htm>, last accessed March 2004.
14. Meetinghouse Data Communications, [http://www.mtghouse.com/MDC\\_8021X\\_White\\_Paper.pdf](http://www.mtghouse.com/MDC_8021X_White_Paper.pdf), last accessed March 2004.
15. Randy Roberts, <http://www.sss-mag.com/ss.html>, last accessed March 2004.
16. Webopedia, [http://www.webopedia.com/TERM/W/Wi\\_Fi.html](http://www.webopedia.com/TERM/W/Wi_Fi.html), last accessed March 2004.
17. NETGEAR, [http://www.netgear.com/products/prod\\_details.php?prodID=173](http://www.netgear.com/products/prod_details.php?prodID=173), last accessed March 2004.
18. OTC Wireless [http://www.otcwireless.com/specs/WiSER2400.IP\\_Data\\_Sheet.pdf](http://www.otcwireless.com/specs/WiSER2400.IP_Data_Sheet.pdf), last accessed April 2004.
19. DPAC Technologies Corporation, [http://www.dpactech.com/wireless\\_dsp.asp](http://www.dpactech.com/wireless_dsp.asp), last accessed April 2004.
20. Quick Start Guide 39L3700-01A, pdf file from DPAC Technologies Corporation, Garden Grove CA, 2003.
21. Eval Kit Users Guide 39L3701-01A, pdf file from DPAC Technologies Corporation, Garden Grove CA, 2003.
22. CadSoft Online, <http://www.cadsoftusa.com/>, last accessed April 2004.
23. Faruk Yildiz, "Implementation of a Human Avatar for the MARG Project in Networked Virtual Environments," MS Thesis, Naval Postgraduate School, Monterey CA, March 2004.

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Fort Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Chairman Code EC  
Department of Electrical and Computer Engineering  
Naval Postgraduate School  
Monterey, California
4. Chairman Code IS  
Department of Information Science  
Naval Postgraduate School  
Monterey, California
5. Professor Xiaoping Yun, Code EC/Yx  
Department of Electrical and Computer Engineering  
Naval Postgraduate School  
Monterey, California
6. Professor David C. Jenn, Code EC/Jn  
Department of Electrical and Computer Engineering  
Naval Postgraduate School  
Monterey, California
7. Professor Eric R. Bachmann  
Department of Computer Science and System Analysis  
Miami University  
Oxford, Ohio
8. Professor Robert B. McGhee, Code CS/Mz  
Department of Computer Science  
Naval Postgraduate School  
Monterey, California
9. Doug McKinney  
McKinney Technology  
Chandler, Arizona

10. Ioannis R. Saliaris  
Athens, Greece